CORRECTNESS, COMPLETENESS, AND CONSISTENCY OF EQUATIONAL DATA TYPE SPECIFICATIONS

von

Dipl.-Inform. Peter Padawitz

Dem Fachbereich 20 Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
vorgelegte Dissertation

Berichter: Prof. Dr. Hartmut Ehrig
Prof. Dr. Dirk Siefkes

Tag der mündlichen Prüfung: 13. Mai 1983

Berlin 1983

D 83

Abstract

The method of the stepwise extension of equational specifications for abstract data types allows us to prove the correctness of a software system specification in parallel to its stepwise design. The whole specification is correct if the semantics of the "base" specification agrees with the data type model and if its extension is "complete" and "consistent" with respect to the basis.

Starting from the correctness notion for parameterized specifications, which was introduced by the ADJ group, we develop proof-theoretical criteria for correctness, completeness and consistency that originate in the calculus of equational logic. These criteria will be refined more and more: On one hand normalization and confluence properties will be included; on the other hand specifications with conditionals, which presume that Boolean expressions are interpreted as in propositional logic, will be treated seperately. At the end the refinement yields conditions, which are decidable or at least decidable relative to the semantics of the base specification. Their decidability results from their characterization by inductively defined predicates.

The hierarchy of proof-theoretical criteria starts with Correctness Thm. 1.15 and Extension Thms. 2.8 and 2.10, which present the main characterizations of the properties that name these theorems. Thm. 7.7 and - for specifications with conditionals - 8.5 yields decidable but rather weak completeness criteria. Completeness Thm. 8.16 combines syntactical and semantical requirements to the specifications and is used when the exclusively syntactical conditions of 7.7 or 8.5 do not hold. Thms. 9.18 and 10.15 as also - for specifications with conditionals - 11.10 and 11.11 state decidable consistency criteria. 10.15 and 11.10 must be referred to whenever the equations of the base specification are not normalizing.

Zusammenfassung

Die Methode der schrittweisen Erweiterung von Gleichungsspezifikationen für abstrakte Datentypen erlaubt es, parallel zum Entwurf eines Softwaresystems dessen Korrektheitsbeweis zu führen: Der Gesamtentwurf ist korrekt, wenn die Semantik der "Basis"-Spezifikation mit dem Modell des Datentyps übereinstimmt und ihre Erweiterung "vollständig" und "konsistent" bezüglich der Basis ist.

Ausgehend von dem von der ADJ-Gruppe geprägten Korrektheitsbegriff für parametrisierte Spezifikationen entwickeln wir auf der Grundlage des Kalküls der Gleichungslogik beweistheoretische Kriterien für Korrektheit, Vollständigkeit und Konsistenz. Diese Kriterien werden zunehmend verfeinert, wobei einerseits Normalisierungs- und Konfluenzeigenschaften gleichungsinduzierter Termersetzungen einbezogen werden und andererseits Spezifikationen mit Konditionalen, die auf der aussagenlogischen Semantik Boolescher Ausdrücke aufbauen, gesondert behandelt werden. Die Verfeinerung der Kriterien endet bei entscheidbaren oder zumindest relativ zur Semantik der Basisspezifikation entscheidbaren Bedingungen. Ihre Entscheidbarkeit folgt aus ihrer Charakterisierung durch induktiv definierte Prädikate.

Von zentraler Bedeutung in dieser Hierarchie beweistheoretischer Kriterien sind zunächst das Korrektheitstheorem 1.15 und die Extensionstheoreme 2.8 und 2.10, die wesentliche Charakterisierungen der die Theoreme benennenden Eigenschaften beinhalten. Der Satz 7.7 bzw. - für Spezifikationen mit Konditionalen - 8.5 liefert entscheidbare Vollständigkeitskriterien, die - wie Beispiele belegen werden - noch recht schwach sind. Das Vollständigkeitstheorem 8.16 verbindet syntaktische und semantische Anforderungen an die Spezifikation und findet häufig dann Anwendung, wenn die ausschließlich syntaktischen Bedingungen von 7.7 bzw. 8.5 nicht gelten. Die Sätze 9.18 und 10.15 bzw. - für Spezifikationen mit Konditionalen - 11.10 und 11.11 geben entscheidbare Konsistenzkriterien an, wobei auf 10.15 bzw. 11.11 dann zurückgegriffen werden muß, wenn die Gleichungen der Basisspezifikation nicht normalisierend sind.

Contents

Preface

Since the mid-sixties two ways have been followed to accomplish correct software. One way is characterized by the concepts of horizontal and vertical program structuring. Horizontal structuring is the decomposition of a problem into subproblems whose program solutions have only a few interfaces where they depend on each other. The notions module (cf. Parnas /54/) and data type (evolving from the "class"-construct of SIMULA; cf. Dahl, Hoare /10/) arise in the context of structured programming. A data type comprehends the collection of at least all data structures with the same initializing, state-changing and inquiry operations. The vertical structuring of software systems by stepwise refinement (cf. Dijkstra /14/, Wirth /65/) divides the way from a problem to its program solution into small steps which should admit to design software and in parallel, to prove its correctness.

The other way has led to several methods to describe formally the meaning (semantics) of programming language constructs. From the theoretical point of view we have to differ the axiomatic from the denotational approach. The axiomatic method is based on Floyd's /21/ and Hoare's /27/ assertion calculus that consists of axioms and deduction rules on assertions <p,s,q> where p and q are logical formulas while s denotes a program. The assertion calculus defines the semantics of the programming language: If an assertion <p,s,q> is derivable, then condition q holds true after s has been executed in a state that satisfied p. On the other hand, the denotational approach proceeds as follows: The memory states of a machine are represented by the set A^X of functions from a set X of addresses to a - usually structured - set A of values. The meaning of a program s is given by a mapping $sem(s): A^X \longrightarrow A^X$ that is inductively defined on the structure of s. If every logical formula p is interpreted in the classical way by a function $sem(p): A^X \longrightarrow \{true, false\}$, then the mapping

h: formulas x programs x formulas $\longrightarrow [A^X \longrightarrow \{\text{true, false}\}]$ with

$$h(p,s,q) = (sem(p) \Longrightarrow sem(q) \circ sem(s))$$

assigns to each assertion $<p,s,q>$ its denotational meaning. ("$\Longrightarrow$" stands for the implication extended to $[A^X \rightarrow \{\text{true,false}\}]$. Thus the denotational method yields a possible model for the assertion calculus. Correctness of the latter with respect to denotational semantics is proved in Manna, Vuillemin /47/. Its completeness presumes that the calculus is "expressive" for the primitive types of the language, i.e. each formula q and each program s need a "weakest precondition" p with $sem(p) \circ sem(s) = sem(q)$ (cf. de Bakker /5/, Wand /63/).

In the mid-seventies both approaches to reliable software: structured programming and mathematical semantics, crossed each other. The method of stepwise refinement has led to specification languages which provide means for structuring the higher levels of software design. Like programming languages, specification languages need a mathematical semantics enabling the software designer to verify an implementation against a specification with the help of an abstraction function from the semantics of the implementation to that of the specification. Papers that initiated the development of specification languages (Hoare /28/, Liskov, Zilles /46/, Guttag /24/, ADJ /1/) have shown that data types are appropriate units for such languages and their semantics. Again, we can distinguish between axiomatic and denotational approaches. The concept of the ADJ group /1/ defines an algebraic specification of a data type as an equational theory presentation SPEC = <S,OP,E>: S is a set of sorts, OP is a set of S-sorted operation symbols, and E denotes a set of equations between terms over OP and S-sorted variables. The semantics of SPEC and thus the data type specified by SPEC is given by any object of the isomorphism class of all initial SPEC-algebras. Analogously to axiomatic programming language semantics, a deduction system determines the semantics of such specifications:

An eminent initial SPEC-algebra is the quotient algebra

$$G_{SPEC} = G_{OP}/\equiv_{SPEC}$$

where G_{OP} is the "free" algebra of ground (i.e. variable-free) OP-terms and $\equiv_{SPEC}$ is the least congruence relation on G_{OP} that contains all ground term instances of E-equations. Two ground terms t,t' are regarded as two representations of the same data structure instance if and only if they are equal modulo $\equiv_{SPEC}$, i.e. if and only if "t = t'" is derivable by the rules of equational calculus from ground term instances of E.

An approach to structuring such specifications vertically was developped by Guttag, Horowitz, Musser /26/, ADJ /1/, Nourani /50/ and extended to a concept that clearly distinguishes between syntax, semantics and correctness by Ehrig, Kreowski, Mahr, Padawitz /20/.

The denotational specification method is characterized by the use of <u>abstract models</u>: The specification of a data type consists of some explicitly defined algebra. An intermediate position between axiomatic and denotational specifications is occupied by <u>final</u> or <u>terminal specifications</u>. Syntactically, they agree with the algebraic specifications sketched above, but their semantics is given by the coarsest quotient of G_{OP} that respects certain primitive types (Giarratana, Gimona, Montanari /22a/, Wand /64/, Kamin /38/, Hornung, Raulefs /29/). Hence the term congruence corresponding to final semantics identifies all ground terms which are "behaviourally equivalent" with respect to the primitive types. Although this congruence can be defined in terms of $\equiv_{SPEC}$, a deduction system that generates it is not known. Equations of final specifications resemble definitions of inquiry functions while - in contrast to algebraic specifications with initial algebra semantics - relations between "generator operations" do not have to be axiomatized.

In the sequel we are exclusively concerned with equational specifications and initial algebra semantics, and we shall develop criteria for their correct horizontal structuring. Moreover, it will be shown that some of these criteria are decidable, which in principle follows from the proof-theoretical nature on initial algebra semantics sketched above.

1. Introduction to algebraic specifications and their correctness

We start with the syntactical notions (1.1 - 1.5), continue with semantics and correctness (1.6 - 1.16) including a characterization theorem for correctness (1.15) and close this chapter with the notion of persistency and its use (1.17 - 1.22). First we collect some

Notations

Inclusions and natural mappings are denoted by inc resp. nat. Let A be a set. id_A and id stand for the identity on A. [a] is the equivalence class of all elements of A equivalent to a with respect to some given equivalence relation on A. A* is the free monoid (set of words) over A. λ and ε denote the empty word. A^+ is $A^*-\{\varepsilon\}$. For all $w \in A^*$ and all functions g with domain A, lg(w) is the length of w, w_i the i-th letter of w and gw the concatenation of $gw_1, \ldots, gw_{lg(w)}$.
Let R be a binary relation. R^{Δ}, R^+ and R* denote the reflexive, transitive resp. reflexive-transitive closure of R. Furthermore, $\langle a,b\rangle^{-1} = \langle b,a\rangle$ and $R^{-1} = \{\langle b,a\rangle / \langle a,b\rangle \in R\}$.
A family of sets is very often identified with their union!

FP(A) denotes the set of finite subsets of A. For each two subsets M and N of A*, M·N is the set of all words $vw \in A^*$ with $v \in M$ and $w \in N$. Multiplication on the set {true, false} means logical conjunction. The 0-ary logical conjunction coincides with the constant "true".

1.1 Definition

An (algebraic) specification SPEC = <S,OP,E> consists of a set S of sorts, a family $OP = \{OP_{w,s}\}_{w \in S^*, s \in S}$ of sets of operation symbols and a set E of OP-equations (see below). <S,OP> is called the signature of SPEC.

Instead of "$\sigma \in OP_{w,s}$" we often write "$\sigma : w \longrightarrow s \in OP$". w and s are the arity resp. sort of σ. lg(w) is called the rank of σ. In the case that w equals λ, σ is a constant, and we write "$\sigma : \longrightarrow s$".
Let $X = \{X_s\}_{s \in S}$ be a fixed family of sets of variables. For each $x \in X_s$ s is the sort of x. $T_{OP} = \{T_{OP,s}\}_{s \in S}$ denotes the family of sets of OP-terms defined inductively by (i), (ii) and (iii), respectively:

(i) for all $s \in S$ $X_s \cup OP_{\lambda,s} \subseteq T_{OP,s}$,
(ii) for all $s \in S$, $w \in S^+$, $\sigma : w \longrightarrow s$ and $t \in T_{OP,w}$ $\sigma t \in T_{OP,s}$,
(iii) for all $n > 0$ and $w \in S^n$

$$T_{OP,w} = \{(t_1, \ldots, t_n)/t_i \in T_{OP,w}\}.$$

Let $w \in S^+$, $s \in S$, $\sigma \in OP_{w,s}$ and $t \in T_{OP,w}$. w is the sort of t, σ is the root of σt and $arg(\sigma t) = t$ are the arguments of σt. We regard σt as a new operation symbol and call it derived from OP. The arity of σt is inductively defined by

$$arity(\sigma t) = \begin{cases} \lambda & \text{if } t = \varepsilon \\ s & \text{if } t \in X \\ arity(t_1) \ldots arity(t_n) & \\ & \text{if } n = lg(w) > 0. \end{cases}$$

op(t) resp. var(t) denotes the set of operation symbols resp. variables of t and size(t) the number of operation symbol and variable occurrences in t.
If $var(t) = \emptyset$, t is called a ground OP-term (tuple). The set of ground term tuples of sort w is denoted by $G_{OP,w}$, and $G_{OP} = \{G_{OP,s}\}_{s \in S}$.
An OP-equation <l,r> of sort $s \in S$ is a pair of OP-terms with sort s. We often write l = r instead of <l,r>.

In examples we use the following syntactical schema for listing the sorts $s_1,\ldots,s_k$, operation symbols $\sigma_1,\ldots,\sigma_m$ and equations $e_1,\ldots,e_n$ of SPEC:

SPEC
  sorts: $s_1,\ldots,s_k$
  opns: $\sigma_1,\ldots,\sigma_m$
  eqns: $e_1,\ldots,e_n$

1.2 Example (bool)

An algebraic specification of truth values and operations of propositional logic reads as follows. The symbols $\wedge, \vee$ and $\rightarrow$ are used in infix notation, and x,y denote variables.

bool

sorts: bool

opns: TRUE, FALSE $: \longrightarrow$ bool
$\neg$: bool $\longrightarrow$ bool
$\wedge, \vee, \rightarrow$: bool bool $\longrightarrow$ bool
IFB: bool bool bool $\longrightarrow$ bool

eqns:

$\neg$TRUE = FALSE	b1
$\neg$FALSE = TRUE	b2
$x \wedge$ TRUE = x	b3
$x \wedge$ FALSE = FALSE	b4
$x \vee$ TRUE = TRUE	b5
$x \vee$ FALSE = x	b6
$x \rightarrow y = (\neg x) \vee y$	b7
IFB(TRUE, x,y) = x	b8
IFB(FALSE,x,y) = y	b9

1.3 Example (nat)

bool is extended by a specification of natural numbers with equality. The symbol "+" after bool forms the componentwise union of bool with additional sorts, operation symbols and equations.

nat = bool +

sorts : nat

opns : O : $\longrightarrow$ nat
S,P: nat $\longrightarrow$ nat
+,$\cdot$: nat nat $\longrightarrow$ nat
EQN : nat nat $\longrightarrow$ bool
IFN : bool nat nat $\longrightarrow$ nat

eqns :

PO = O	n1
PSx = x	n2
x+O = x	n3
x+Sy = S(x+y)	n4
x·O = O	n5
x·Sy = (x·y)+x	n6
EQN(O,O) = TRUE	n7
EQN(O,Sx) = FALSE	n8
EQN(Sx,O) = FALSE	n9
EQN(Sx,Sy) = EQN(x,y)	n10
IFN(TRUE,x,y) = x	n11
IFN(FALSE,x,y) = y	n12

The stepwise extension of specifications by sorts, operations and/or equations as in Example 1.3 is one way of structuring algebraic specifications. It gets more powerful when it is combined with the concept of parameterization developped in ADJ/2/, Ehrich /15/ and Ehrig /16/.

1.4 Definition

A parameterized specification PAR = <PSPEC,SPEC> consists of two specifications PSPEC = <PS,POP,PE> and SPEC = <S,OP,E> with $PS \subseteq S$, $POP \subseteq OP$ and $PE \subseteq E$. PSPEC and SPEC are called the (formal) parameter resp. target specification of PAR.

1.5 Example (array)

A parameterized specification of unbounded arrays (partial functions with finite domain) is given by PAR = <entry,array> where

entry = nat +

sorts : entry

opns : UNDEF : $\longrightarrow$ entry
EQE : entry entry $\longrightarrow$ bool
IFE : bool entry entry $\longrightarrow$ entry

eqns : IFE(TRUE,x,y) = x e1
IFE (FALSE,x,y) = y e2

and

array = entry +

sorts : array

opns : NEW : $\longrightarrow$ array
PUT : array nat entry $\longrightarrow$ array
IFA : bool array array $\longrightarrow$ array

eqns : PUT(NEW,n,UNDEF) = NEW a1
PUT(PUT(a,n,x),m,y) =

IFA(EQN(n,m),
    PUT(a,m,y),
    PUT(PUT(a,m,y),n,x)) a2

IFA (TRUE,a,b) = a a3

IFA (FALSE,a,b) = b a4

x,y,n,m,a and b are variables.

Since algebraic specifications are presentations of a theory, it is natural to say that the data type of a specification SPEC corresponds to the class of models of the theory presented by SPEC. As we only allow equational axioms in SPEC, the model class is a variety, i.e. a class of "equationally defined" algebras.

1.6 Definition

Let SPEC = $<S,OP,E>$ be a specification and SIG its signature. A SIG - algebra A consists of a carrier set A_s for all $s \in S$ and an operation $\sigma_A : A_s \longrightarrow A_s$ for each $\sigma \in OP_{w,s}$ where $A_w = A_{w_1} x \ldots x A_{w_n}$ and n = lg(w).
If $w = \lambda$, we obtain $\sigma_A \in A_s$.
Alg(SIG) denotes the class of SIG-algebras.
The families T_{OP} of OP-terms and G_{OP} of ground OP-terms become $<S,OP>$ - algebras by defining

$$T_{OP,s} = G_{OP,s} = \emptyset \quad \text{for all } s \in S\text{-sort}(OP),$$

$$\sigma_T(t) = \sigma t \quad \text{for all } \sigma \in OP_{w,s} \text{ and } t \in T_{OP,w}$$

and

$$\sigma_G(t) = \sigma t \quad \text{for all } \sigma \in OP_{w,s} \text{ and } t \in G_{OP,w}.$$

Let A and B be SIG- algebras. The family $h = \{h_s : A_s \longrightarrow B_s\}_{s \in S}$ of functions is called a SIG - homomorphism, written $h:A \rightarrow B$, if for all

$\sigma \in OP_{w,s}$ $h_s \circ \sigma_A = \sigma_B \circ h_w$ where $h_\lambda = \emptyset$, $h_w = h_{w1} \times \cdots \times h_{wn}$ and $n = lg(w)$. A family $f = \{f_s : X_s \longrightarrow A_s\}_{s \in S}$ of functions is called an <u>assignment to A</u>. <u>Z(A)</u> denotes the set of assignments to A. For all $f \in Z(A)$ there is a unique SIG-homomorphism $\underline{f^*}$: $T_{OP} \longrightarrow A$ with $f = f^* \circ inc$. f^* is called the <u>term evaluation in A w.r.t. f</u> and will mostly be identified with f!
Moreover there is a unique SIG - homomorphism $\underline{eval}_A : G_{OP} \rightarrow A$ called <u>term evaluation</u>. $eval_A(t)$ is abbreviated by $\underline{t}_A$.

A <u>satisfies</u> an equation $\langle l,r \rangle \in E$ if for all $f \in Z(A)$ $fl = fr$. A is a <u>SPEC-algebra</u> if A satisfies all $\langle l,r \rangle \in E$. <u>Alg(SPEC)</u> denotes the class of SPEC-algebras. A is an <u>initial SPEC-algebra</u> if for all $B \in Alg(SPEC)$ there is a unique SIG - homomorphism $h_B : A \longrightarrow B$.

It is known from universal algebra and recalled by ADJ/1/ that an initial SPEC-algebra exists for each specification SPEC and that the class of initial SPEC-algebras is isomorphism-closed and only contains isomorphic objects. ADJ/1/ call this class the <u>initial semantics of SPEC</u>.
This notion of data type semantics has been thoroughly motivated in ADJ/1/ and papers based upon it. In our introductory remarks on axiomatic and denotational definitions of semantics we have mentioned the proof-theoretical nature of initial semantics:

1.7 <u>Definition and Theorem</u> (ADJ/1/, Thm.6)

Let SPEC = $\langle S,OP,E \rangle$ be a specification. The least OP-congruence relation on G_{OP} which contains all pairs $\langle fl,fr \rangle$ with $\langle l,r \rangle \in E$ and $f \in Z(G_{OP})$ is called the <u>SPEC-congruence</u> and is denoted by $\equiv_{SPEC}$.
Then the "quotient term algebra" $G_{SPEC} = G_{OP}/\equiv_{SPEC}$ is an initial SPEC-algebra. □

From a model-theoretical point of view, the initial SPEC-algebras are the "maximal" elements in the class of minimal SPEC-algebras: Each OP-equation that holds true in some initial SPEC-algebra is satisfied by all minimal SPEC-algebras, i.e. by all SPEC-algebras with surjective term evaluation.

Up to know we have described the way from specifications to their models, the algebras. In software design we are rather interested in the opposite direction, namely how to get from a given data type model to its specification. In order to cope with this problem, Ehrig, Kreowski, Padawitz /18/ have introduced the following correctness notion for specifications:

1.8 Definition

Given a signature MSIG and an MSIG-algebra A, a specification SPEC with MSIG$\subseteq$SPEC componentwise is correct w.r.t. A if $U_{<MSIG,SPEC>}(G_{SPEC}) \cong A$ where $U_{<MSIG,SPEC>}$ denotes the forgetful functor from Alg(SPEC) to Alg(MSIG).

Using 1.8 we are forced to formalize a data type model as an algebra A of some "model signature" MSIG. A correct specification of A will often have a signature that properly includes MSIG: Some MSIG-operations may only be specifiable with the help of additional, previously specified operations or even sorts.

ADJ /2/,/3/ and Ehrig /16/ have then adapted the identities

data types = algebras

and

SPEC-semantics = initial SPEC-algebras

to the more general situation of a parameterized specification PAR = <PSPEC,SPEC> (cf. 1.4): For a given class K of PSPEC-algebras the initial semantics of PAR w.r.t K is defined as the composition $F_{PAR} \circ IN$ of the inclusion functor $IN: K \longrightarrow Alg(PSPEC)$ and the free functor

$$F_{PAR}: Alg(PSPEC) \longrightarrow Alg(SPEC).$$

We assume that the reader is familiar with the category-theoretical term "functor" and the "adjointness" between free and forgetful functors (cf. e.g. Arbib, Manes /4/, chapter 7).

For any pair $<P1,P2>$ of signatures or specifications with $P1 \subseteq P2$ componentwise $F_{<P1,P2>}$ denotes the free functor from Alg(P1) to Alg(P2), while $U_{<P1,P2>}$ denotes its (forgetful) right adjoint.

This context immediately yields the following characterization of $F_{PAR} \circ IN$:

1.9 Proposition

Let PAR = <PSPEC,SPEC> be a parameterized specification with signatures PSIG and SIG of PSPEC resp. SPEC. Let K be a class of PSPEC-algebras and IN be the inclusion functor from K to Alg(PSPEC). Then $F = F_{PAR} \circ IN$ has the following property (*), and for all mappings $F : K \longrightarrow Alg(SPEC)$ that satisfy (*) and for all $A \in K$, $F(A) \cong F_{PAR}(A)$.

(*) For all $A \in K$ there is a PSIG-homomorphism $\eta_A: A \longrightarrow F_{PAR}(A)$ such that for all $B \in Alg(SPEC)$ each PSIG-homomorphism $h: A \longrightarrow B$ uniquely extends to a SIG-homomorphism $h^+: F_{PAR}(A) \rightarrow B$ with $h^+ \circ \eta_A = h$. □

For the following two reasons it is obvious to call $F_{PAR} \circ IN$ an initial semantics of PAR: Firstly, in the case that PSPEC is empty we have $Alg(PSPEC)=\{\emptyset\}$. Hence $F_{PAR} \circ IN$ may be identified with

$F_{PAR}(\emptyset)$ which, by proposition 1.9, is an initial SPEC-algebra. Secondly, for each $A \in K$ the <u>unit morphism</u> η_A given by 1.9 (*) is initial in the comma category $[A, U_{PAR}]$ whose objects h are the PSIG-homomorphisms from A to arbitrary SPEC-algebras and whose morphisms $g \in Mor(h,h')$ are the SIG-homomorphisms with $g \circ h = h'$ (Arbib, Manes /4/, p. 113).

The correctness notion of 1.8 is extended to parameterized specifications as follows:

1.10 <u>Definition</u>

Let MPSIG and MSIG be two "model" signatures with $MPSIG \subseteq MSIG$ componentwise, and let F be a mapping from a class MK of MPSIG-algebras to Alg(MSIG).
A pair <u><PAR,K></u> consisting of

(i) a parameterized specification PAR = <PSPEC,SPEC> with $MPSIG \subseteq PSPEC$ and $MSIG \subseteq SPEC$ componentwise

and

(ii) a class K of PSPEC-algebras with $U_{<MPSIG,PSPEC>}(K) = MK$

is called <u>correct w.r.t. F</u> if the following diagram commutes up to isomorphism:

$$\begin{tikzcd}
 & MK \arrow[rr, "F"] & & Alg(MSIG) \\
 & K \arrow[u, "U_{<MPSIG,PSPEC>}"] \arrow[r, "IN"'] & Alg(PSPEC) \arrow[r, "F_{PAR}"'] & Alg(SPEC) \arrow[u, "U_{<MSIG,SPEC>}"']
\end{tikzcd}$$

1.11 Example (array, cf. 1.5)

Let PSIG and SIG be the signatures of entry resp. array. Let K be the class of all PSIG-algebras A with the following properties:

(i) for all $m,n \in A_{\underline{nat}}$

$$EQN_A(m,n) = \begin{cases} TRUE_A & \text{if } m = n \\ FALSE_A & \text{otherwise,} \end{cases}$$

(ii) $A_{\underline{bool}} = \{TRUE_A, FALSE_A\}$ and $TRUE_A \neq FALSE$.

A mapping $F: K \longrightarrow Alg(SIG)$ is given by

$(FA)_s = A_s$ for all $s \in PS$,

$(FA)_{\underline{array}} = \{a: A_{\underline{nat}} \longrightarrow A_{\underline{entry}} /\ an \neq UNDEF_A$ for at most finitely many $n \in A_{\underline{nat}}\}$,

$\sigma_{FA} = \sigma_A$ for all $\sigma \in POP$,

$NEW_{FA}(n) = UNDEF_A$,

$PUT_{FA}(a,n,k) = \lambda i.\text{if } i=n \text{ then } k \text{ else } ai$,

$IFA_{FA}(p,a,b) = \text{if } (p=TRUE_A) \text{ then } a \text{ else } b$
for all $p \in A_{\underline{bool}}$, $n \in A_{\underline{nat}}$, $k \in A_{\underline{entry}}$
and $a,b \in (FA)_{\underline{array}}$.

Since all $A \in K$ satisfy the equations of entry, K is a class of entry-algebras. Let PAR = <entry, array>. Using Prop. 1.9 we show that <u><PAR,K> is correct w.r.t. F:</u>

Since for all $A \in K$ FA satisfies E, we can regard F as a mapping from K to Alg(SPEC).
Let $A \in K$. We choose $\eta_{A,s} = id$ for all $s \in PS$.
Let $B \in Alg(array)$ and $h : A \longrightarrow B$ be a PSIG-homomorphism.

Let f be an arbitrary function which maps each finite nonempty subset M of $A_{\underline{nat}}$ to some element of M. For all $s \in PS$ set $h_s^+ = h_s$ and define $h^+_{\underline{array}}$ inductively on /dom(a)/ as follows where $dom(a) = \{n \in A_{\underline{nat}} / an \neq UNDEF_A\}$:

(iii) $h^+_{\underline{array}}(NEW_{FA}) = NEW_B$,

(iv) $h^+_{\underline{array}}(PUT_{FA}(a,n,e))$

$= PUT_B(h^+_{\underline{array}}(a), h_{\underline{nat}}(n), h_{\underline{entry}}(e))$

where $an = UNDEF_A \neq e$

and $n = f(dom(PUT_{FA}(a,n,e)))$.

The compatibility of h^+ with NEW follows from (iii). Hence h^+ is a SIG-homomorphism if for all $a,a' \in (FA)_{\underline{array}}$, $n \in A_{\underline{nat}}$, $e \in A_{\underline{entry}}$ and $p \in A_{\underline{bool}}$

(1) $h^+(PUT_{FA}(a,n,e)) = PUT_B(h^+a, hn, he)$,

(2) $h^+(IFA_{FA}(p,a,a')) = IFA_B(hp, h^+a, h^+a')$.

We prove (1) by induction on /dom(a)/:

Let $a = NEW_{FA}$. $e = UNDEF_A$ implies

$h^+(PUT_{FA}(a,n,e)) = h^+(NEW_{FA}) = NEW_B =$

$= PUT_B(NEW_B, hn, UNDEF_B)$

$= PUT_B(h^+NEW_{FA}, hn, he)$.

$e \neq UNDEF_A$ implies

$h^+(PUT_{FA}(a,n,e)) = PUT_B(h^+a, hn, he)$.

Let $a \neq NEW_{FA}$. If $n \in dom(a)$, then we have for some a_o with $PUT_{FA}(a_o, n, an) = a$ and $a_o n = UNDEF_A$ by induction hypothesis

$h^+(PUT_{FA}(a_o, n, an)) = PUT_B(h^+a_o, hn, han)$

and

$h^+(PUT_{FA}(a_o, n, e)) = PUT_B(h^+a_o, hn, he)$.

(i) implies $EQN_B(hn,hn) = hEQN_A(n,n)$

$= hTRUE_A = TRUE_B$ so that

$h^+(PUT_{FA}(a,n,e)) = h^+(PUT_{FA}(PUT_{FA}(a_o, n, an)n, e))$

$= h^+(PUT_{FA}(a_o, n, e)) = PUT_B(h^+a_o, hn, he)$

$= IFA_B(EQN_B(hn,hn),$

$PUT_B(h^+a_o, hn, he),$

$PUT_B(PUT_B(h^+a_o, hn, he), hn, han))$

$= PUT_B(PUT_B(h^+a_o, hn, han), hn, he)$

$= PUT_B(h^+(PUT_{FA}(a_o, n, an)), hn, he)$

$= PUT_B(h^+a, hn, he)$.

Let $n \notin dom(a)$ and $n_o = f(dom(PUT_{FA}(a,n,e)))$.
If $n \notin dom(a)$, then $n \neq n_o$, and we get for all a_o with $PUT_{FA}(a_o, n_o, an_o) = a$ and $a_o n = UNDEF_A$ by induction hypothesis

$$h^+(PUT_{FA}(a_o,n_o,an_o)) = PUT_B(h^+a, hn_o, h(an_o))$$

and

$$h^+(PUT_{FA}(a_o,n,e)) = PUT_B(h^+a_o, hn, he).$$

(i) implies $EQN_B(hn_o, hn) = h(EQN_A(n_o,n)) = hFALSE_A = FALSE_B$ so that

$$\begin{aligned}
h^+(PUT_{FA}(a,n,e)) &= h^+(PUT_{FA}(PUT_{FA}(a_o,n_o,an_o),n,e)) \\
&= h^+(PUT_{FA}(PUT_{FA}(a_o,n,e),n_o,an_o)) \\
&= PUT_B(h^+(PUT_{FA}(a_o,n,e)),hn_o,han_o) \text{ by (iv)} \\
&= PUT_B(PUT_B(h^+a_o,hn,he),hn_o,han_o) \\
&= IFA_B(EQN_B(hn_o,hn), \\
&\qquad PUT_B(h^+a_o,hn,he), \\
&\qquad PUT_B(PUT_B(h^+a_o,hn,he),hn_o,han_o)) \\
&= PUT_B(PUT_B(h^+a_o,hn_o,han_o),hn,he) \\
&= PUT_B(h^+(PUT_{FA}(a_o,n_o,an_o)),hn,he) \\
&= PUT_B(h^+a,hn,he).
\end{aligned}$$

If $n \notin dom(a)$, then $n = n_o$, and we obtain (1) by (iv).

Thus we have shown (1) for all cases. (2) holds true as follows:

$$\begin{aligned}
& h^+(IFA_{FA}(p,a,a')) \\
&= h^+(\text{if } p=TRUE_A \text{ then } a \text{ else } a') \\
&= \text{if } p=TRUE_A \text{ then } h^+a \text{ else } h^+a' \\
&= \text{if } p=TRUE_A \text{ then } IFA_B(TRUE_B,h^+a,h^+a') \\
&\qquad\qquad\quad \text{else } IFA_B(FALSE_B,h^+a,h^+a') \\
&= IFA_B(hp,h^+a,h^+a') \quad \text{by (ii).}
\end{aligned}$$

Uniqueness of h^+ with respect to $h^+ \circ \eta_A = h$ is an immediate consequence of the definition of h^+.
□

The correctness proof of Example 1.11 is completely model-theoretic. In contrast to that we are now going to generalize the quotient term construction of initial semantics (cf. 1.7) to parameterized specifications. This provides the basis for showing the correctness of parameterized specifications by proof-theoretical (term rewriting) methods.

1.12 <u>Definition</u>

Let PAR = <PSPEC, SPEC> be a parameterized specification with PSPEC = <PS, POP, PE> and SPEC = <S, OP, E>. Let A be a PSPEC-algebra. Assume for the moment that for all $s \in PS$ the set X_s of variables agrees with the carrier set A_s of A. Hence we have the term evaluation $id_A{}^*: T_{POP} \longrightarrow A$ which extends the assignment

$$id_A = \{id_{A_s}: A_s \longrightarrow A_s\}_{s \in PS}:$$

$$\begin{tikzcd}
A \arrow[rr, "inc"] \arrow[dr, "id_A"'] & {} \arrow[d, phantom, "="] & T_{POP} \arrow[dl, "id_A{}^*"] \\
& A &
\end{tikzcd}$$

The <u>equational diagram of A, $\Delta(A)$</u>, is given by all $\langle t,t'\rangle \in T^2_{POP}$ with $id_A{}^*t = id_A{}^*t'$. Intuitively,

$\Delta(A)$ is the set of all POP-equations satisfied by A.

1.13 <u>Theorem</u>

Let PAR = <PSPEC, SPEC> be a parameterized specification with PSPEC = <PS, POP, PE> and SPEC = <S, OP, E>. Then for all PSPEC-algebras A

$$F_{PAR}(A) \cong G_{SPEC(A)}$$

where $\underline{SPEC(A)} = \langle S, OP(A), E(A)\rangle$ and for all $s \in S$, $w \in S^+$

$$OP(A)_{\lambda,s} = OP_{\lambda,s} \cup A_s,$$
$$OP(A)_{w,s} = OP_{w,s},$$
$$E(A) = E \cup \Delta(A).$$

Proof:

Let $A \in Alg(PSPEC)$ and set $\eta_A = A \xrightarrow{inc} G_{OP(A)} \xrightarrow{nat} G_{SPEC(A)}$.
By Prop. 1.9 we have to show that for all SPEC-algebras B and $\langle PS, POP\rangle$-homomorphisms $h: A \longrightarrow B$ there is a unique $\langle S, OP\rangle$-homomorphism $h^+: G_{SPEC(A)} \longrightarrow B$ with $h^+ \circ \eta_A = h$.
B becomes an $\langle S, OP(A)\rangle$-algebra if we set $a_B = ha$ for all $a \in A$.
Suppose that B satisfies $\Delta(A)$. Then B is a SPEC(A)-algebra and, by initiality of $G_{SPEC(A)}$, we obtain a unique $\langle S, OP(A)\rangle$-homomorphism h^+ from $G_{SPEC(A)}$ to B. On the other hand, let for all $s \in PS$ $X_s = A_s$ and for all $s \in S\text{-}PS$ $X_s = \emptyset$. Then the $\langle PS, POP\rangle$-homomorphism $h: A \longrightarrow B$ can be regarded as an assignment, and we get the term evaluation $h^*: T_{OP} \longrightarrow B$ w.r.t. B as the unique $\langle S, OP\rangle$-homomorphism satisfying

$$\begin{array}{ccc} A & \xrightarrow{\quad inc \quad} & T_{OP} \\ & h \searrow \quad = \quad \swarrow h^* & \\ & B & \end{array}$$

Since $T_{OP} = G_{OP(A)}$, h^* agrees with $h^+ \circ nat$.
Hence $h^+ \circ \eta_A = h$.
Vice versa, $h' \circ \eta_A = h$ for some $\langle S, OP\rangle$-homomorphism $h': G_{SPEC(A)} \longrightarrow B$ implies $h' \circ nat = h^*$ by uniqueness of h^*. Again, since $T_{OP} = G_{OP(A)}$, h^* is compatible with A as a set of constants, and thus

h^* is $\langle S, OP(A)\rangle$-homomorphic. Since nat is an $\langle S, OP(A)\rangle$-epimorphism, $h' \circ nat = h^*$ implies that h' is $\langle S, OP(A)\rangle$-homomorphic. Therefore, $h' = h^+$. It remains to show that $\Delta(A)$ holds true in B. Let $\langle t, t'\rangle \in \Delta(A)$. The term evaluation

$eval_B : G_{POP(A)} \longrightarrow B$ satisfies

$$\begin{tikzcd}
A \arrow[rr, "inc"] \arrow[dr, "h"'] \arrow[rr, phantom, shift right=4ex, "="] & & G_{POP(A)} \arrow[dl, "eval_B"] \\
& B &
\end{tikzcd}$$

Hence $eval_B = h \circ id_A{}^*$ (see the diagram in Def. 1.12), and we obtain $eval_B t = eval_B t'$. $\square$

Note that $G_{SPEC(A)}$ is different from the "quotient term algebra with parameter variables" used by Ganzinger /22/. This algebra characterizes the free functor going from $Alg(\langle S, \emptyset\rangle)$, the category of S-sorted sets, to Alg(SPEC).

1.14 <u>Definition and Proposition</u> (cf. 1.7)

Let $SPEC = \langle S, OP, E\rangle$ be a specification. The least OP-congruence relation on T_{OP} which contains all pairs $\langle fl, fr\rangle$ with $\langle l, r\rangle \in E$ and $f \in Z(T_{OP})$ is called the <u>SPEC-congruence with variables</u> and is denoted by $\approx_{SPEC}$. Then the "quotient term algebra with variables" $T_{SPEC} = T_{OP}/\approx_{SPEC}$ satisfies

$$F_{NEW}(X) \cong T_{SPEC}$$

where $NEW = \langle\langle S, \emptyset\rangle, SPEC\rangle$.

Proof:

For all $A \in Alg(SPEC)$ and $h \in Z(A)$ an OP-homomorphism $h^+ : T_{SPEC} \longrightarrow A$ is uniquely defined by the following diagram:

$$\begin{tikzcd}
X \arrow[r, "inc"] \arrow[dr, "h"'] & T_{OP} \arrow[r, "nat"] \arrow[d, "h^*"] \arrow[dl, phantom, "=", near start] \arrow[dr, phantom, "=", near start] & T_{SPEC} \arrow[dl, "h^+"] \\
& A &
\end{tikzcd}$$

Prop. 1.9, applied to PAR = NEW, yields
$T_{SPEC} \cong F_{NEW}(X)$. $\square$

The characterization of F_{PAR} in Thm. 1.13 provides a correctness criterion for parameterized specifications:

1.15 Correctness Theorem

Let F, PAR and K be as in Def. 1.10, MPSIG = =<MPS,MPOP> and MSIG =<MS,MOP>. <PAR,K> is correct w.r.t. F iff for all $A \in MK$

(i) FA is extendable to a SPEC(A)-algebra,

(ii) the term evaluation $eval_{FA} : G_{MOP(A)} \longrightarrow FA$ has a right-inverse g such that for all $w \in MS^*$, $s \in MS$, $\sigma \in MOP(A)_{w,s}$ and $a \in (FA)_w$
$$\sigma g_w(a) \equiv_{SPEC(A)} g_s \circ \sigma_{FA}(a).$$

If $U_{<MPSIG,MSIG>}(FA) = A$, then it is sufficient to set g_s = inc for all $s \in MPS$ and to postulate (ii) only for all $\sigma \in$ MOP-MPOP.

(ii) generalizes the representation condition for canonical term algebras defined in ADJ /1/, Thm. 9.

<u>Proof</u>:

Let $A \in MK$, $MOP(A)$ be defined analogous to $OP(A)$ in 1.13, $MSIG(A) = <MS, MOP(A)>$ and $SIG(A)$ be the signature of $SPEC(A)$.
By Thm. 1.13 we have to show that for all $A \in K$ FA is isomorphic to $\hat{G}_{SPEC(A)} := U_{<MSIG,SPEC>} G_{SPEC(A)}$.
Let $\hat{G}_{SPEC(A)} \xrightarrow{h} FA$. Initiality of $G_{MOP(A)}$ implies

$$\begin{tikzcd} & G_{MOP(A)} \arrow[dl, "nat"'] \arrow[dr, "eval_{FA}"] \arrow[d, phantom, "="] & \\ \hat{G}_{SPEC(A)} \arrow[rr, "h"'] & & FA \end{tikzcd} \tag{1}$$

Hence $eval_{FA}$ is surjective and has a right-inverse g. Thus $h \circ nat \circ g = eval_{FA} \circ g = id$ is an $MSIG(A)$-homomorphism so that, by injectivity of h, $nat \circ g$ is $MSIG(A)$-homomorphic, too. Initiality of $G_{MOP(A)}$ therefore implies $nat = nat \circ g \circ eval_{FA}$.
Hence for
all $w \in MS^*$, $s \in MS$, $\sigma \in MOP(A)_{w,s}$ and $a \in (FA)_w$

$$\begin{aligned} \sigma g_w(a) &\equiv_{SPEC(A)} g_s \circ eval_{FA}(\sigma g_w(a)) \\ &= g_s \circ \sigma_{FA} \circ eval_{FA} \circ g_w(a) \\ &= g_s \circ \sigma_{FA}(a). \end{aligned}$$

Vice versa, let FA be a $SPEC(A)$-algebra and g a right-inverse of $eval_{FA} : G_{MOP(A)} \longrightarrow FA$ with (ii).
By induction on size(t), (ii) implies for all $t \in G_{MOP(A)}$

$$t \equiv_{SPEC(A)} g \circ eval_{FA}(t). \tag{2}$$

Since $G_{SPEC(A)}$ and $G_{OP(A)}$ are initial in $Alg(SPEC(A))$ and $Alg(SIG(A))$, respectively, we obtain a $SIG(A)$-homomorphism h' with

$$\begin{tikzcd}
& G_{OP(A)} \arrow[dl, "nat"'] \arrow[dr, "eval'"] \arrow[d, phantom, "="] & \\
G_{SPEC(A)} \arrow[rr, "h'"'] & & FA
\end{tikzcd}$$

For $h = U_{<SIG,SPEC>}(h')$ diagram (1) is again commutative. Since $eval_{FA}$ has a right-inverse, it is surjective and thus h is surjective, too. The injectivity of h follows from (2): For all $t,t' \in G_{MOP(A)}$ $h \circ nat(t) = h \circ nat(t')$ implies

$nat(t) = nat \circ g \circ eval_{FA}(t) = nat \circ g \circ h \circ nat(t) =$
$= nat \circ g \circ h \circ nat(t') = nat \circ g \circ eval_{FA}(t') = nat(t')$.

Hence $G_{SPEC(A)}$ is isomorphic to FA.
Now suppose that we have $U_{PAR}(FA) = A$.
Then for all $s \in MPS$ and $a \in A_s$ $eval_{FA}(a) = a$ so that $g_s = inc$ is a right-inverse of $eval_{FA,s}$.
Moreover, for all $w \in MPS^*$, $s \in MPS$, $\sigma \in MPOP(A)_s$ and $a \in (FA)_w$

$$\sigma g_w(a) = \sigma a \equiv_{SPEC(A)} \sigma_A(a) = g\sigma_A(a) = g\sigma_{FA}(a)$$

because $<\sigma a, \sigma(a)> \in \triangle(A)$. $\square$

The following example presents an alternative correctness proof of <<entry,array>,K> w.r.t. F with K and F being defined as in 1.11. We use the correctness criterion of Thm. 1.15. There is a strong similarity between this proof and that one given in 1.11. The role of the SIG-homomorphism $h^+ : FA \longrightarrow B$ in 1.11 will be taken over by the "choice function" $g : FA \longrightarrow G_{OP(A)}$, the right-inverse of $eval_{FA}$ (cf. 1.15(ii)). Equational rewriting of ground OP(A)-terms to "canonical" ones, i.e. to g-images (1.15 (ii)) corresponds to the SIG-homomorphism property of h^+: For all $w \in S^*$, $s \in S$, $\sigma \in OP_{w,s}$ and $a \in (FA)_w$ the equation

$$\sigma_B h^+_w(a) = h^+_s \circ \sigma_{FA}(a)$$

will be replaced by the congruence

$$\sigma g_w(a) \equiv_{SPEC(A)} g_s \circ \sigma_{FA}(a).$$

Although in both cases the proof steps are nearly the same, we put more emphasis on the second one because it illustrates that a significant part of correctness proofs consists of term rewriting.

1.16 Example (array, cf. 1.11)

Let PSIG and SIG be the signatures of entry resp. array (cf. 1.5). Let $K \subseteq Alg(PSPEC)$ and $F: K \longrightarrow Alg(SPEC)$ be defined as in 1.11, and let PAR = <entry,array>.
Using Thm. 1.15 we show that <u><PAR, K> is correct w.r.t. F:</u>
Since for all $A \in K$ FA is a SPEC(A)-algebra with $a_{FA} = a$ for all $a \in A$, 1.15(i) holds true.
Let $A \in K$ and f be an arbitrary function which maps each finite nonempty subset M of $A_{\underline{nat}}$ to some element of M. For all $s \in PS$ set g_s = inc and define $g_{\underline{array}}$ inductively on /dom(A)/ as follows where $dom(a) = \{u \in A_{\underline{nat}} / an \neq UNDEF_A\}$:

(i) $g_{\underline{array}}(NEW_{FA}) = NEW$,

(ii) $g_{\underline{array}}(PUT_{FA}(a,n,e)) = PUT(g_{\underline{array}}(a),n,e)$
where $an = UNDEF_A \neq e$
and $n = f(dom(PUT_{FA}(a,n,e)))$.

A first induction on /dom(a)/ yields the fact that $g_{\underline{array}}$ is a right-inverse of $eval_{FA,\underline{array}}$. By Thm. 1.15, it remains to show that for all $a,a' \in (FA)_{\underline{array}}$, $n \in A_{\underline{nat}}$, $e \in A_{\underline{entry}}$ and $p \in A_{\underline{bool}}$

(1) $NEW \equiv_{SPEC(A)} g(NEW_{FA})$,

(2) $PUT(ga,n,e) \equiv_{SPEC(A)} g(PUT_{FA}(a,n,e))$,

(3) $IFA(p,ga,ga') \equiv_{SPEC(A)} g(IFA_{FA}(p,a,a'))$.

(1) follows from (i). We prove (2) by induction on /dom(a)/.
Let $a = NEW_{FA}$. $e = UNDEF_A$ implies

$$PUT(ga,n,e) \equiv_{SPEC(A)} NEW = g(NEW_{FA}).$$

$e \neq UNDEF_A$ implies

$$PUT(ga,n,e) = g(PUT_{FA}(a,n,e)).$$

Let $a \neq NEW_{FA}$. If $n \in dom(a)$, then we obtain for some a_o with $PUT_{FA}(a_o,n,an) = a$ and $a_o n = UNDEF_A$ by induction hypothesis

$$PUT(ga_o,n,an) \equiv_{SPEC(A)} g(PUT_{FA}(a_o,n,an))$$

and

$$PUT(ga_o,n,e) \equiv_{SPEC(A)} g(PUT_{FA}(a_o,n,e)).$$

By definition of K (cf. 1.11), $<EQN(n,n), TRUE>$ belongs to $\Delta(A)$ so that

$$\begin{aligned}
PUT(ga,n,e) &= PUT(g(PUT_{FA}(a_o,n,an)),n,e)\\
&\equiv_{SPEC(A)} PUT(PUT(ga_o,n,an),n,e)\\
&\equiv_{SPEC(A)} IFA(EQN(n,n),\\
&\qquad\qquad PUT(ga_o,n,e),\\
&\qquad\qquad PUT(PUT(ga_o,n,e),n,an)\\
&\equiv_{SPEC(A)} PUT(ga_o,n,e)\\
&\equiv_{SPEC(A)} g(PUT_{FA}(a_o,n,e)) = g(PUT_{FA}(a,n,e)).
\end{aligned}$$

Let $n \notin dom(a)$ and $n_o = f(dom(PUT_{FA}(a,n,e)))$. If $n_o \in dom(a)$, then $n \neq n_o$, and we get for all a_o with $PUT_{FA}(a_o,n_o,an_o) = a$ and $a_o n_o = UNDEF_A$ by induction hypothesis

$$PUT(ga_o,n_o,an_o) \equiv_{SPEC(A)} g(PUT_{FA}(a_o,n_o,an_o))$$

and

$$PUT(ga_o,n,e) \equiv_{SPEC(A)} g(PUT_{FA}(a_o,n,e)).$$

Again by definition of K, A satisfies $<EQN(n_o,n),FALSE>$ so that

$$\begin{aligned}
PUT(ga,n,e) &= PUT(g(PUT_{FA}(a_o,n_o,an_o)),n,e)\\
&\equiv_{SPEC(A)} PUT(PUT(ga_o,n_o,an_o),n,e)\\
&\equiv_{SPEC(A)} IFA(EQN(n_o,n),\\
&\qquad\qquad PUT(ga_o,n,e)\\
&\qquad\qquad PUT(PUT(ga_o,n,e),n_o,an_o))
\end{aligned}$$

$$\equiv_{SPEC(A)} PUT(PUT(ga_o,n,e),n_o,an_o)$$
$$\equiv_{SPEC(A)} PUT(g(PUT_{FA}(a_o,n,e)),n_o,an_o)$$
$$\overset{(*)}{=} g(PUT_{FA}(PUT_{FA}(a_o,n,e),n_o,an_o))$$
$$= g(PUT_{FA}(a,n,e))$$

where equation (*) follows from (ii) because

$$PUT_{FA}(a_o,n,e)(n_o) = a_o n_o = UNDEF_A \neq an_o.$$

If $n_o \notin dom(a)$, then $n = n_o$ and $e \neq UNDEF_A$, and we conclude

$$PUT(ga,n,e) = g(PUT_{FA}(a,n,e))$$

from (ii).

Thus we have shown (2) for all cases.

By definition of K, $A_{\underline{bool}} = \{TRUE_A, FALSE_A\}$. Hence (3) is a consequence of

$$IFA(TRUE_A,ga,ga') \equiv_{SPEC(A)} ga$$

and

$$IFA(FALSE_A,ga,ga') \equiv_{SPEC(A)} ga'. \quad \square$$

We close this section by a short glance at the way how a parameterized specification PAR = <PSPEC,SPEC> gets actualized. Replacing the formal parameter PSPEC in SPEC by an actual parameter ACTUAL yields a new specification VALUE such that $F_{PAR}(G_{ACTUAL})$ should agree with G_{VALUE}, i.e. the semantics of PAR assigns the semantics of ACTUAL to the semantics of VALUE. The connection between the formal and the actual parameter is established by a specification morphism:

1.17 Definition (ADJ/3/,4.1,4.2)

Let SPEC = <S,OP,E> and SPEC' = <S',OP',E'> be specifications. A specification morphism $<f,g>:SPEC \rightarrow SPEC'$ consists of a mapping $f:S \longrightarrow S'$ and a family $g = \{g_{w,s}: OP_{w,s} \longrightarrow OP'_{f^*w,fs}\}_{w \in S^*, s \in S}$ of mappings such that the "translation" gE of E is a subset of E'.

1.18 <u>Definition</u> (ADJ/3/, 4.3)

Let PAR = <PSPEC,SPEC> with PSPEC = <PS,POP,PE> and SPEC = <S,OP,E> be a parameterized specification, ACTUAL = <PS',POP',PE'> be a specification and <f,g>: PSPEC $\longrightarrow$ ACTUAL a specification morphism. The <u>actualized specification</u> VALUE = <S',OP',E'> is defined by

$$S' = PS' \cup S,$$

$$OP' = POP' \cup \{\sigma' : f'^*w \longrightarrow f's \,/\, \sigma \in OP_{w,s} - POP\},$$

and

$$E' = PE' \cup g'(E-PE)$$

where for all $s \in S$

$$f's = \text{if } s \in PS \text{ then } fs \text{ else } s$$

and for all $\sigma \in OP$

$$g'\sigma = \text{if } \sigma \in POP \text{ then } g\sigma \text{ else } \sigma'.$$

Thus <f',g'> is a specification morphism from SPEC to VALUE.

The coincidence of formal and actual semantics is made precise by the notion of compatibility:

1.19 <u>Definition</u> (ADJ/3/, 4.3)

Given PAR,ACTUAL,<f,g>, VALUE and <f',g'> as in Def. 1.18, ACTUAL is called <u>compatible with PAR</u> if

$$F_{PAR}U_{\langle f,g\rangle}(G_{ACTUAL}) \cong U_{\langle f',g'\rangle}(G_{VALUE})$$

where $U_{\langle f,g\rangle}$ and $U_{\langle f',g'\rangle}$ denote the forgetful functors from Alg(ACTUAL) resp. Alg(VALUE) to Alg(PSPEC) resp. Alg(SPEC).

$$\begin{tikzcd}[column sep=huge]
\mathrm{ACTUAL} \arrow[r, mapsto, "\text{syntactical actualization}"] \arrow[d, mapsto, "\text{semantics}"'] & \mathrm{VALUE} \arrow[d, mapsto, "\text{semantics}"] \\
U_{\langle f,g\rangle}(G_{\mathrm{ACTUAL}}) \arrow[r, mapsto, "F_{\mathrm{PAR}}", "\text{semantical actualization}"'] & U_{\langle f',g'\rangle}(G_{\mathrm{VALUE}})
\end{tikzcd}$$

The next theorem provides a criterion for compatibility. The crucial property of parameterized specifications which admit compatible actualizations is persistency:

1.20 Definition (cf. ADJ/2/,p.721)

A parameterized specification PAR = <PSPEC,SPEC> is persistent w.r.t. $K \subseteq \mathrm{Alg(PSPEC)}$ if for all $A \in K$

$$U_{\mathrm{PAR}} F_{\mathrm{PAR}}(A) \cong A.$$

1.21 Theorem

Given PAR, ACTUAL and <f,g> as in Def. 1.18, ACTUAL is compatible with PAR if PAR is persistent w.r.t. $\{U_{\langle f,g\rangle}(G_{\mathrm{ACTUAL}})\}$. □

Theorem 1.21 is a special case of Thm. 4.2 in Ehrig /16/ where the parameter of PAR is again a parameterized specification. But it is more general than the corresponding Thm. 5.2 in ADJ/3/ because we do not require persistency w.r.t. all PSPEC-algebras but only w.r.t. a subclass of Alg(PSPEC) which contains $U_{\langle f,g\rangle}(G_{\mathrm{ACTUAL}})$.

1.22 <u>Example</u> (array, cf. 1.11)

nat is compatible with PAR = <entry,array> (cf. 1.3 and 1.5): A specification morphism <f,g> from entry to nat is defined by

$$fs = s \text{ for all } s\in\{\underline{bool},\underline{nat}\},$$
$$f(\underline{entry}) = \underline{nat},$$
$$g(\sigma) = \sigma \text{ for all } \sigma\in\{TRUE,FALSE,\neg,\wedge,\vee,\rightarrow, IFB,O,S,EQN\},$$
$$g(EQN) = EQN \text{ and } g(IFE) = IFN.$$

Since $U_{\langle f,g\rangle}(G_{nat})$ belongs to the class K of entry-algebras defined in Example 1.11, we have

$$U_{\langle PSIG,SIG\rangle}F(U_{\langle f,g\rangle}(G_{nat})) = U_{\langle f,g\rangle}(G_{nat})$$

for the mapping $F:K \longrightarrow Alg(SIG)$ given in 1.11.
By correctness of <PAR,K> w.r.t.F, we obtain

$$F(U_{\langle f,g\rangle}(G_{nat})) \cong U_{\langle SIG,SPEC\rangle}\circ F_{PAR}(U_{f,g}(G_{nat}))$$

and thus

$$U_{\langle PSIG,SPEC\rangle}\circ F_{PAR}(U_{\langle f,g\rangle}(G_{nat})) \cong U_{\langle f,g\rangle}(G_{nat}).$$

Since PSIG is the signature of PSPEC, PAR is persistent w.r.t. $\{U_{\langle f,g\rangle}(G_{nat})\}$, and compatibility of nat with PAR follows from Thm. 1.21. □

Besides being a criterion for compatibility of actual parameters, persistency is also a useful restriction to "algebraic implementations" (see Preface) because persistent implementations are closed under composition (Ehrig,Kreowski,Mahr,Padawitz/20/, 6.4 + 7.3).

In the next chapter persistency will turn out as a special case of correct extensionality.

2. Correct extensions of specifications

Now we come to that field in the theory of algebraic specifications which we shall treat with term rewriting methods in the next chapters. It is the stepwise extension of parameterized specifications and its impact on semantics and correctness.
We take as the basis the following

General assumption

BPAR = <PSPEC, BSPEC> and PAR = <PSPEC, SPEC> are two parameterized specifications such that PSPEC = <PS, POP, PE> is componentwise included in BSPEC = <BS, BOP, BE> and the latter is componentwise contained in SPEC = <S, OP, E>. S is finite, and for all $w \in S^*$ and $s \in S$ $OP_{w,s}$ is finite.
$\underline{X} = \{X_s\}_{s \in S}$ is a fixed family of variables such that for all $s \in S$ X_s is infinite. The subfamily $\{X_s\}_{s \in BS}$ is abbreviated by $\underline{BX}$.
K is a nonempty class of PSPEC-algebras. $\underline{T}$, $\underline{BT}$, $\underline{G}$ and $\underline{BG}$ denote the term algebras T_{OP}, T_{BOP}, G_{OP} and G_{BOP}, respectively. Correspondingly, for all $A \in K$ $\underline{T(A)}$, $\underline{BT(A)}$, $\underline{G(A)}$ and $\underline{BG(A)}$ are abbreviations for $T_{OP(A)}$, $T_{BOP(A)}$, $G_{OP(A)}$ and $G_{BOP(A)}$, respectively.
For all $A \in K$, $s \in S$ and $s' \in BS$ $G(A)_s$ as well as $BG(A)_{s'}$ are supposed to be nonempty.
For all $M \in \{T, G\} \cup \{T(A)/A \in K\} \cup \{G(A)/A \in K\}$, $\underline{BZ(M)}$ denotes the set of all assignments $f \in Z(M)$ with $f(BX) \subseteq BM$, e.g.

$$BZ(T(A)) = \{f \in Z(T(A))/f(BX) \subseteq BT(A)\}.$$

We first give a criterion for the correctness of PAR in the case that BPAR is already correct. Hence it admits the joint stepwise development of a specification and its correctness proof.

2.1 Correctness Propagation Theorem

Let $MPSIG = \langle MPS, MPOP \rangle$, $MBSIG = \langle MBS, MBOP \rangle$ and $MSIG = \langle MS, MOP \rangle$ be "model" signatures with $MPSIG \subseteq MBSIG \subseteq MSIG$, $MBSIG \subseteq MBSPEC$ and $MSIG \subseteq MSPEC$ componentwise and let F be a mapping from a class MK of MPSIG-algebras to Alg(MSIG). Suppose that $U_{\langle MPSIG, PSPEC \rangle}(K) = MK$.
Let $\langle BPAR, K \rangle$ be correct w.r.t. $BF := U_{\langle MBSIG, MSIG \rangle} \circ F$.
Hence some right-inverse g of $eval_{BFA}: G_{MBOP(A)} \longrightarrow BFA$ satisfies (cf. 1.15)

$$\sigma g_w(a) \equiv_{BSPEC(A)} g_s \circ \sigma_{BFA}(a)$$

for all $w \in MBS^*$, $s \in MBS$, $\sigma \in MBOP(A)_{w,s}$ and $a \in (BFA)_w$.
$\langle PAR, K \rangle$ is correct w.r.t. F if for all $A \in MK$

(i) FA is extendable to a SPEC(A)-algebra,

(ii) for all $s \in MS\text{-}MBS$ there is a right-inverse g'_s of $eval_{FA,s}: G_{MOP(A),s} \longrightarrow FA_s$ such that for all $w \in MS^*$, $s \in MS$, $\sigma \in MOP_{w,s}\text{-}MBOP$ and $a \in (FA)_w$

$$\sigma g'_w(a) \equiv_{SPEC(A)} g'_s \circ \sigma_{FA}(a) \qquad (*)$$

where $g'_s = g_s$ for all $s \in MBS$.

Proof:

Let $U = U_{\langle MBSIG, MSIG \rangle}$. We have

$$\begin{tikzcd}
G_{MBOP(A)} \arrow[r, "inc"] \arrow[d, bend right, "eval_{BFA}"'] & UG_{MOP(A)} \arrow[d, bend left, "Ueval_{FA}"] \\
BFA \arrow[u, "g"'] \arrow[r, phantom, "="] \arrow[ur, phantom, "="] & UFA \arrow[u, "Ug'"]
\end{tikzcd}$$

By (ii) and the assumption about g, g' is a right-inverse of $eval_{FA}$, and (*) holds for all $\sigma \in MOP(A)$. Thus correctness of $\langle PAR, K \rangle$ w.r.t. F follows from Thm. 1.15. □

2.2 Example (array1)

Let BPAR = <entry,array> (cf. 1.5),

array1 = array +

opns: GET: $\underline{\text{array}}\ \underline{\text{nat}} \longrightarrow \underline{\text{entry}}$

eqns: GET(NEW,n) = UNDEF a5

GET(PUT(a,n,x),m) =
= IFE(EQN(n,m),
x,
GET(a,m)) a6

and PAR = <entry,array1>. Let K be that class of entry-algebras which was defined in 1.11. If BSIG and SIG denote the signatures of array resp. array1, the mapping F : K $\longrightarrow$ Alg(BSIG) given in 1.11 is extended to F : K $\longrightarrow$ Alg(SIG) by

$$GET_{FA}(a,n) = a(n)$$

for all $a \in (FA)_{\underline{array}}$ and $n \in A_{\underline{nat}}$.
Using Thm. 2.1 we show that <u><PAR,K> is correct w.r.t. F</u>:

In 1.11 and 1.16 we have proved the correctness of BPAR w.r.t BF = $U_{<BSIG,SIG>} \circ F$.
Since for all $A \in K$ FA is a SPEC(A)-algebra with $a_{FA} = a$ for all $a \in A$, 2.1 (i) holds true. Thus it remains to show that for all $A \in K$, $a \in (FA)_{\underline{array}}$ and $n \in A_{\underline{nat}}$

$$GET(ga,n) \equiv_{SPEC(A)} a(n) \qquad (*)$$

where g is the right-inverse of $eval_{BFA}$ defined in 1.16. We prove (*) by induction on /dom(a)/ (cf. 1.16):
/dom(a)/ = 0 implies a = NEW_{FA} so that

$$GET(ga,n) = GET(NEW,n) \equiv_{SPEC} UNDEF$$
$$\equiv_{SPEC(A)} UNDEF_A = an.$$

If /dom(a)/ > 0, then there are a',n',e with $PUT_{FA}(a',n',e) = a$ where $a'n' = UNDEF_A \neq e$ and $n' = f(dom(a))$ (cf. 1.16). Hence /dom(a')/ < /dom(a)/ and by induction hypothesis,

$$GET(ga,n) = GET(PUT(ga',n',e),n)$$

$$
\begin{aligned}
&\equiv_{SPEC} IFE(EQN(n',n),e,GET(ga',n)) \\
&\equiv_{SPEC(A)} \begin{cases} IFE(TRUE,e,GET(ga',n)) & \text{if } n' = n \\ IFE(FALSE,e,GET(ga',n)) & \text{otherwise} \end{cases} \\
&\equiv_{PSPEC} \begin{cases} e = an' = an & \text{if } n' = n \\ GET(ga',n) \equiv_{SPEC(A)} a'n = an & \text{otherwise.} \end{cases} \quad \square
\end{aligned}
$$

In many cases we are not given a "full model" $F : MK \longrightarrow Alg(MSIG)$ but only its basic part $BF : MK \longrightarrow Alg(MBSIG)$. Then we are not interested in an explicit definition of F but only in the fact that BF is extendable to some F such that PAR becomes correct w.r.t. F. PAR should have a "model" F whose "MBSIG-part" agrees with BF, in other words: PAR should be logically consistent with respect to BF:

2.3 <u>Definition</u>

Let MPSIG, MBSIG and MK be as in Thm. 2.1. Let SIG be the signature of SPEC and MEXT = <MBSIG,SIG>. PAR is <u>logically consistent w.r.t. $BF : MK \longrightarrow Alg(MBSIG)$</u> if there is

$F : MK \longrightarrow Alg(SIG)$ with $U_{MEXT} \circ F = BF$

such that <PAR,K> is correct w.r.t. F.

Let <u>EXT</u> = <BSPEC,SPEC>. The diagram of Def. 1.10 extends to our present situation as follows:

$$\begin{tikzcd}
\mathrm{MK} \arrow[rr, "BF"] \arrow[rrr, bend left, "F"] & & \mathrm{Alg(MBSIG)} & \mathrm{Alg(SIG)} \arrow[l, "U_{MEXT}"'] \\
\mathrm{K} \arrow[u, "U"] \arrow[r, "IN"] \arrow[dr, "IN"'] & \mathrm{Alg(PSPEC)} \arrow[r, "F_{BPAR}"] & \mathrm{Alg(BSPEC)} \arrow[u, "U"] & \mathrm{Alg(SPEC)} \arrow[l, "U_{EXT}"'] \arrow[u, "U"'] \arrow[ul, phantom, "="] \\
& \mathrm{Alg(PSPEC)} \arrow[urr, "F_{PAR}"'] & &
\end{tikzcd}$$

The indices of upgoing forgetful functors are omitted. Using the notation of 1.10 we now have MSIG = SIG.

Since we assume U(K) = MK, diagram chasing yields the following

2.4 Lemma

Let <BPAR,K> be correct w.r.t. BF. PAR is logically consistent w.r.t. BF iff for all $A \in K$

$$UU_{EXT}F_{PAR}(A) \cong UF_{BPAR}(A). \quad \square$$

2.4 motivates the

2.5 Definition

PAR is a correct extension of <BPAR,K> if for all $A \in K$

$$U_{EXT}F_{PAR}(A) \cong F_{BPAR}(A).$$

If PSPEC = $<\emptyset,\emptyset,\emptyset>$ and K = $\{\emptyset\}$, PAR is a correct extension of <BPAR,K> iff $U_{EXT}(G_{SPEC}) \cong G_{BSPEC}$.

Hence, in the unparameterized case, correct extensions are exactly extensions in the sense of ADJ/1/, Def. 9.

Lemma 2.4 immediately implies

2.6 Proposition

Let <BPAR,K> be correct w.r.t. BF.

(i) PAR is logically consistent w.r.t. BF if PAR is a correct extension of <BPAR,K>.

(ii) PAR is a correct extension of <BPAR,K> if PAR is logically consistent w.r.t. BF and MBSIG is the signature of BSPEC. □

The rest of this chapter deals with necessary and/or sufficient conditions for correct extensions. Two of them, semantical completeness and semantical consistency, resemble the characterization of correctness in Thms. 1.15 and 2.1.
The others are "proof-theoretical" in that they are properties of the BSPEC(A)- and SPEC(A)-congruences (cf. 1.13 and 1.7).

2.7 Definition

PAR is <u>(syntactically) complete w.r.t. <BPAR,K></u> if for all $A \in K$, $s \in BS$ and $t \in G(A)_s$ there is $t' \in BG(A)$ with $t \equiv_{SPEC(A)} t'$.

PAR is <u>semantically complete w.r.t. <BPAR,K></u> if for all $A \in K$ $F_{BPAR}(A)$ is extendable to an OP-algebra A' and there is a function $g_{A'}: A' \to G(A)$

such that

(i) for all $s \in BS$ resp. $s \in S-BS$ $g_{A',s}$ is a right-inverse of the term evaluation $eval: BG(A) \longrightarrow F_{BPAR}(A)$ resp. $eval': G(A) \longrightarrow F_{PAR}(A)$,

(ii) for all $w \in S^*, s \in S, \sigma \in OP_{w,s}-BOP$ and $a \in A'_w$

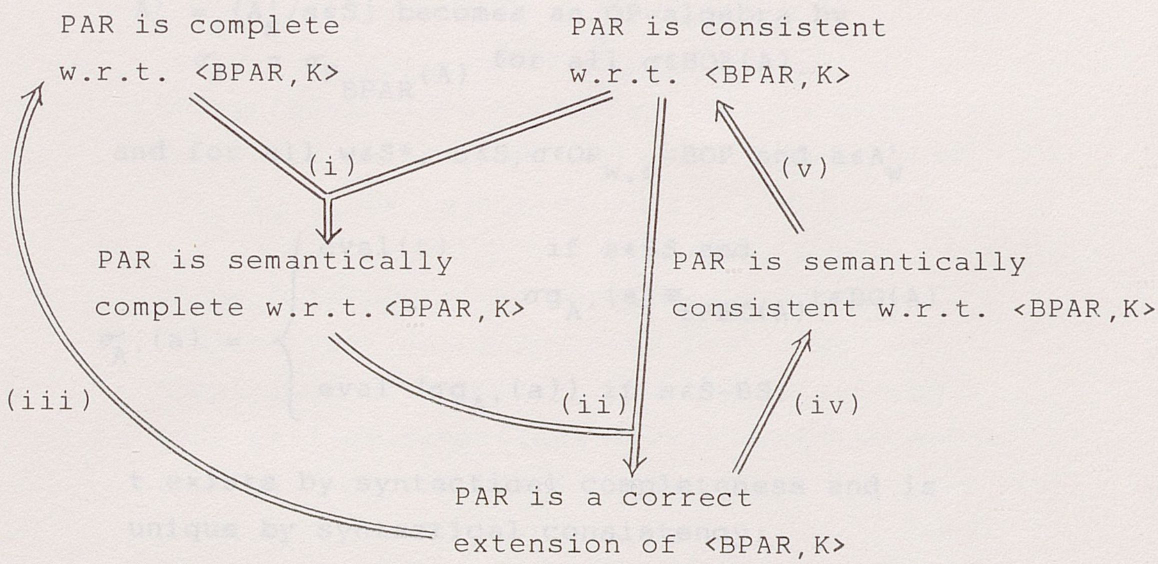
$$\sigma g_{A'}(a) \equiv_{SPEC(A)} g_{A'} \circ \sigma_{A'}(a).$$

PAR is <u>(syntactically) consistent w.r.t. <BPAR,K></u> if for all $A \in K$ and $t,t' \in BG(A)$ $t \equiv_{SPEC(A)} t'$ implies $t \equiv_{BSPEC(A)} t'$.

PAR is <u>semantically consistent w.r.t. <BPAR,K></u> if for all $A \in K$ $F_{BPAR}(A)$ is extendable to a SPEC-algebra A'.

2.8 Extension Theorem

The properties defined in 2.5 and 2.6 are related as follows:



<u>Proof</u>:

(i) Let $A \in K$ and the BOP(A)- resp. OP(A)-homomorphisms nat, eval, h resp. nat', eval', h' be definded by

$$\begin{tikzcd} & BG(A) \arrow[dl, "nat"'] \arrow[dr, "eval"] \arrow[d, phantom, "="] & \\ G_{BSPEC(A)} \arrow[rr, "h"] & & F_{BPAR}(A) \end{tikzcd} \quad \text{resp.} \quad \begin{tikzcd} & G(A) \arrow[dl, "nat'"'] \arrow[dr, "eval'"] \arrow[d, phantom, "="] & \\ G_{SPEC(A)} \arrow[rr, "h'"] & & F_{PAR}(A) \end{tikzcd}$$

By Thm. 1.13, h and h' are bijective. Since $nat \circ h = eval$, eval is surjective and thus has a right-inverse g. Hence $h \circ nat \circ g = eval \circ g = id$ is a BOP(A)-homomorphism so that, by injectivity of h, $nat \circ g$ is BOP(A)-homomorphic, too. Therefore, initiality of BG(A) implies

$$nat \circ g \circ eval = nat. \qquad (*)$$

Analogously, we obtain a right-inverse g' of eval' with

$$nat' \circ g' \circ eval' = nat' . \qquad (**)$$

For all $s \in BS$ let $A'_s = F_{BPAR}(A)_s$ and $g_{A',s} = g_s$.
For all $s \in S\text{-}BS$ let $A'_s = F_{PAR}(A)_s$ and $g_{A',s} = g'_s$.
$A' = \{A'_s / s \in S\}$ becomes an OP-algebra by

$$\sigma_{A'} = \sigma_{F_{BPAR}(A)} \text{ for all } \sigma \in BOP(A),$$

and for all $w \in S^*$, $s \in S$, $\sigma \in OP_{w,s}$-BOP and $a \in A'_w$

$$\sigma_{A'}(a) = \begin{cases} eval(t) & \text{if } s \in BS \text{ and } \sigma g_{A'}(a) \equiv_{SPEC(A)} t \in BG(A) \\ eval'(\sigma g_{A'}(a)) & \text{if } s \in S\text{-}BS. \end{cases}$$

t exists by syntactical completeness and is unique by syntactical consistency:
$t \equiv_{SPEC(A)} t'$ for $t, t' \in BG(A)$ implies $t \equiv_{BSPEC(A)} t'$ and thus $eval(t) = eval(t')$.
Finally, we have to check (ii). Let $w \in S^*$, $s \in S$, $\sigma \in OP_{w,s}$-BOP and $a \in A'_w$. $s \in BS$ implies

$nat'(\sigma g_{A'} a) = nat'(t) = nat(t) = nat \circ g \circ eval(t)$
$= nat \circ g \circ \sigma_{A'}(a) = nat' \circ g \circ \sigma_{A'}(a) = nat' \circ g_{A'} \circ \sigma_{A'}(a)$

for some $t \in BG(A)$. If $s \in S\text{-}BS$, then

$nat'(\sigma g_{A'} a) = nat' \circ g' \circ eval'(\sigma g_{A'} a) =$
$= nat' \circ g' \circ \sigma_{A'}(a) = nat' \circ g_{A'} \circ \sigma_{A'}(a).$

(ii) Let $A \in K$ and nat, eval, h, nat', eval', h' as in the diagrams above. Let $g = \{g_{A',s}\}_{s \in BS}$. Semantical completeness implies $eval \circ g = id$ and $eval_s \circ g_{A',s} = id$ for all $s \in S\text{-}BS$.

Hence $h \circ nat \circ g = eval \circ g = id$ and
$h'_s \circ nat'_s \circ g_{A',s} = eval'_s \circ g_{A',s} = id$
for all $s \in S\text{-}BS$ yields the injectivity of
$nat \circ g$ and $nat'_s \circ \sigma_{A',s}$, $s \in S\text{-}BS$.

Now look at the following diagram where the U's denote forgetful functors:

$$\begin{tikzcd}
F_{BPAR}(A) \arrow[d, "g"'] & = & UA' \arrow[d, "Ug_{A'}"] \\
BG(A) \arrow[rr, "inc"] \arrow[d, "nat"'] & & UG(A) \arrow[d, "Unat'"] \\
G_{BSPEC(A)} \arrow[rr, "init"] & & UG_{SPEC(A)}
\end{tikzcd}$$

By syntactical consistency, init is injective. Hence we conclude from the diagram that $nat'_s \circ g_{A',s}$ is injective also for all $s \in BS$.

Since h is BOP(A)-monomorphic, $h \circ nat \circ g = id$ implies that $nat \circ g$ and thus $Unat' \circ Ug_{A'}$ are compatible with BOP(A). Together with condition (ii) of semantical completeness we infer that $nat' \circ g_{A'}$ is OP(A)-homomorphic.

Hence $nat' \circ g_{A'}$ is OP(A)-monomorphic so that A' is a SPEC(A)-algebra, and we obtain a unique OP(A)-homomorphism $init' : G_{SPEC(A)} \longrightarrow A'$.
By initiality of $G_{SPEC(A)}$,
$nat' \circ g_{A'} \circ init' = id$ so that $nat' \circ g_{A'}$ is surjective and thus an isomorphism between A' and $G_{SPEC(A)}$.
Hence, by Thm. 1.13,

$U_{EXT} F_{PAR}(A) \cong U_{EXT} G_{SPEC(A)} \cong UA' = F_{BPAR}(A)$.

Therefore, PAR is a correct extension of $\langle BPAR, K \rangle$.

(iii) Let $A \in K$. Since $U_{EXT} F_{PAR}(A) \cong F_{BPAR}(A)$, Thm. 1.13 implies $G_{BSPEC(A)} \cong U_{EXT}G_{SPEC(A)}$. Hence PAR is syntactically complete w.r.t. <BPAR,K>.

(iv) Let $A \in K$. Using the BOP(A)-isomorphism $h: U_{EXT}F_{PAR}(A) \longrightarrow F_{BPAR}(A)$, which exists by assumption, we extend $F_{BPAR}(A)$ to an OP-algebra A' as follows:

- for all $s \in S\text{-}BS$ $\quad A'_s = h(F_{PAR}(A)_s)$,
- for all $w \in S^*$, $s \in S$, $\sigma \in OP_{w,s}\text{-}BOP$ and $a \in A'_w$ $\sigma_{A'}(a) = h\sigma_{F_{PAR}(A)}(h^{-1}a)$.

Thus h becomes an OP(A)-isomorphism from $F_{PAR}(A)$ to A' so that A' satisfies E.

(v) Let $A \in K$. We get an OP(A)-homomorphism $h: G_{SPEC(A)} \longrightarrow A'$ such that

$$\begin{tikzcd}
G_{BSPEC(A)} \arrow[r, "init"] \arrow[d, "\wr"'] \arrow[dr, phantom, "="] & U_{EXT}G_{SPEC(A)} \arrow[d, "Uh"] \\
F_{BPAR}(A) \arrow[r, "id"'] & UA'
\end{tikzcd}$$

Hence init is injective, and thus PAR is syntactically consistent w.r.t. <BPAR,K>. □

In the sequel we investigate the case where K consists of all PSPEC-algebras. A with $A_s \neq \emptyset$ for all $s \in PS$. Completeness and consistency conditions for this case were already formulated in Ganzinger /22/. Especially, the proof of Thm. 2.10 (iii) below is an adaption of the proof of Thm. 6 in Ganzinger /22/ to our framework.

For the rest of this chapter assume that <u>for all $s \in S\text{-}PS$ the set X_s of variables is empty</u>.
Moreover, for all $t \in T$, $f \in Z(T)$ and $Y \subseteq X$

$t[x \longleftarrow fx \ / x \in Y]$ denotes $f't$
where $f' \in Z(T)$ is defined by

$$f'z = \begin{cases} fz & \text{if } z \in Y \\ z & \text{otherwise.} \end{cases}$$

2.9 Definition (cf. 1.14)

PAR is X-complete w.r.t. BPAR if for all $s \in BS$ and $t \in T_s$ there is $t' \in BT$ with $t \approx_{SPEC} t'$.
PAR is X-consistent w.r.t. BPAR if for all $t, t' \in BT$ $t \approx_{SPEC} t'$ implies $t \approx_{BSPEC} t'$.

2.10 Extension Theorem

Let K be the class of all PSPEC-algebras with $A_s \neq \emptyset$ for all $s \in PS$. Then we obtain the following implications in addition to those given in Thm. 2.8:

$$\begin{tikzcd}
\begin{array}{c}\text{PAR is a correct}\\ \text{extension of } \langle BPAR, K\rangle\end{array} \arrow[d, Rightarrow, "\text{(i)}"] & & \\
\begin{array}{c}\text{PAR is X-complete}\\ \text{w.r.t. BPAR}\end{array} \arrow[d, Rightarrow, "\text{(ii)}"'] \arrow[drr, Rightarrow, "\text{(iii)}"] & & \begin{array}{c}\text{PAR is X-consistent}\\ \text{w.r.t. BPAR}\end{array} \arrow[d, Rightarrow] \\
\begin{array}{c}\text{PAR is complete}\\ \text{w.r.t. } \langle BPAR, K\rangle\end{array} & & \begin{array}{c}\text{PAR is consistent}\\ \text{w.r.t. } \langle BPAR, K\rangle\end{array} \arrow[u, Rightarrow, bend right, "\text{(iv)}"']
\end{tikzcd}$$

Proof:
Let $PNEW = \langle\langle PS, \emptyset\rangle, PSPEC\rangle$, $BNEW = \langle\langle PS, \emptyset\rangle, BSPEC\rangle$

and NEW = <<PS,$\emptyset$>,SPEC>. Since the composition of two free functors is again free, Prop. 1.14 implies

$$T_{BSPEC} \cong F_{BNEW}(X) \cong F_{BPAR}F_{PNEW}(X) \cong F_{BPAR}(T_{PSPEC}) \qquad (*)$$

and analogously,

$$T_{SPEC} \cong F_{PAR}(T_{PSPEC}). \qquad (**)$$

(i) By assumption,

$U_{EXT}F_{PAR}(T_{PSPEC}) \cong F_{BPAR}(T_{PSPEC})$.

Hence, (*) and (**) imply

$U_{EXT}(T_{SPEC}) \cong T_{BSPEC}$

so that PAR is X-complete w.r.t. BPAR.

(ii) Let $A \in K$, $s \in BS$ and $t \in T(A)$. Then there are $u \in T$ and $f \in Z(A)$ such that $t = u[x \longleftarrow fx/x \in X]$. By assumption, there is $u' \in BT$ with $u \approx_{SPEC} u'$. Hence $t \equiv_{SPEC(A)} fu'$.

(iii) Let $A \in K$ and $t,t' \in BT(A)$ with $t \equiv_{SPEC(A)} t'$. Then there are a least number n and $t_1,\ldots,t_n \in T(A)$ such that $t_1 = t$, $t_n = t'$ and for all $1 \leq i < n$

$$t_i \xleftrightarrow[E(A)]{} t_{i+1}$$

where $\xleftrightarrow[E(A)]{}$ denotes the symmetric closure of the E(A)-reduction relation (cf. 4.1). Moreover, there are $f \in Z(A)$ and $u_i \in T$, $1 \leq i \leq n$, such that f is injective on $\bigcup_{i=1}^{n} var(u_i)$ and $fu_i = t_i$. X-completeness of PAR implies $u_i \approx_{SPEC} u_i'$ for some $u_i' \in BT$.

Next we show that for all $1 \leq i \leq n$

$$fu_i' \equiv_{BSPEC(A)} fu_{i+1}'.$$

If $t_i \xleftrightarrow[E]{} t_{i+1}$, then $u_i' \approx_{SPEC} u_i \xleftrightarrow[E]{} u_{i+1} \approx_{SPEC} u_{i+1}'$ so that

$u_i' \approx_{BSPEC} u_{i+1}'$ by X-consistency of PAR.

Therefore,

$$fu_i' \equiv_{BSPEC(A)} fu_{i+1}'.$$

If $t_i \xleftrightarrow[\Delta(A)]{} t_{i+1}$, then there are $u \in T, z \in var(u)$ and $v, v' \in T_{POP}$ with $u_i = u[z \leftarrow v]$, $u_{i+1} = u[z \leftarrow v']$ and

$$\langle fv, fv' \rangle \in \Delta(A).$$

X-completeness of PAR implies $u \approx_{SPEC} u'$ for some $u' \in BT$. Hence

$$u_i' \approx_{SPEC} u_i = u[z \leftarrow v] \approx_{SPEC} u'[z \leftarrow v]$$

and thus, by X-consistency of PAR, $u_i' \approx_{BSPEC} u'[z \leftarrow v]$. Analogously, $u_{i+1}' \approx_{BSPEC} u'[z \leftarrow v']$. Therefore,

$$fu_i' \equiv_{BSPEC(A)} f(u'[z \leftarrow v]) = (fu')[z \leftarrow fv] \equiv_{BSPEC(A)} (fu')[z \leftarrow fv'] = f(u'[z \leftarrow v']) \equiv_{BSPEC(A)} fu_{i+1}'.$$

Hence $fu_1' \equiv_{BSPEC(A)} fu_n'$. Since $u_1, u_n \in BT$, X-consistency of PAR implies $u_1 \approx_{BSPEC} u_1'$ and $u_n \approx_{BSPEC} u_n'$, which finally gives us

$$t = fu_1 \equiv_{BSPEC(A)} fu_1' \equiv_{BSPEC(A)} fu_n' \equiv_{BSPEC(A)} fu_n = t'.$$

(iv) Let $A = T_{PSPEC}$. By assumption, the unique BOP(A)-homomorphism $h: G_{BSPEC(A)} \longrightarrow G_{SPEC(A)}$ is injective. By Thm. 1.13 and (*),(**), we obtain

$$G_{BSPEC(A)} \cong F_{BPAR}(A) \cong T_{BSPEC}$$

and

$$G_{SPEC(A)} \cong F_{PAR}(A) \cong T_{SPEC}.$$

Hence there is a BOP-monomorphism from T_{BSPEC} to T_{SPEC}, which implies that PAR is X-consistent w.r.t. BPAR. □

X-completeness requires that every OP-term can be reduced to some BOP-term. The following two lemmata allow us to confine the proof of this property to a certain subset of T.

Let <u>OP'</u> $= BOP \cup \bigcup_{s \in S-BS} OP_s$ and <u>OP"</u> $= \bigcup_{s \in PS} OP_s$.

2.12 Lemma

PAR is X-complete w.r.t. BPAR iff for all $w \in S^*, s \in S$, $\sigma \in OP_{w,s} - OP'$ and $t \in T_{OP'-OP'',w}$ there is $t' \in BT$ with $\sigma t \approx_{SPEC} t'$.

Proof:

Let $t \in \bigcup_{s \in BS} T_s$. The existence of $t' \in BT$ with $t \approx_{SPEC} t'$ is shown by induction on the number $n(t)$ of $(OP-OP')$-symbols in t.

If $n(t) = 0$, then $t \in BT$. Otherwise t has a minimal subterm $u = \sigma u'$ with $\sigma \in OP-OP'$ and $u' \in T_{OP'}$. Let $v_1, \dots, v_m$ be the maximal subterms of u with $sort(v_i) \in PS$. Then there are $v' \in T_{OP'-OP''}$ and $f \in Z(T)$ with $fv' = u$. Let $v = \sigma v'$.
Hence by assumption, $v \approx_{SPEC} v''$ for some $v'' \in BT$.
Moreover, there are $t_o \in T$ and $z \in X$ with $t_o[z \leftarrow u] = t$.
By induction hypothesis, we obtain

$$t = t_o[z \leftarrow u] = t_o[z \leftarrow fv] \approx_{SPEC} t_o[z \leftarrow fv''] \approx_{SPEC} t'$$

for some $t' \in BT$. □

Derived operators (cf. ADJ/1/,p.99) suggest the following

2.13 Definition

SPEC is <u>derived from BSPEC</u> if E-BE consists of exactly one equation

$$\sigma(x_1, \dots, x_n) = t$$

for each $\sigma \in OP-OP'$ where $x_1, \dots, x_n$ are distinct variables - not necessarily in X - and t' is a BOP-term.

2.14 Theorem

If SPEC is derived from BSPEC, then PAR is a correct extension of <BPAR,K> where K is the class of all PSPEC-algebras with $A_s \neq \emptyset$ for all $s \in PS$.

Proof:

Let $A \in K$ and SPEC be derived from BSPEC.
Lemma 2.11 implies that PAR is X-complete w.r.t. BPAR:

Let $\sigma t \in T$ has $\sigma \in OP-OP'$ and $t \in T^*_{OP'-OP''}$. By assumption, there are distinct variables $x_1, \ldots, x_n$ and $u \in BT$ with

$$\sigma t = \sigma(x_1, \ldots, x_n)[x_i \longleftarrow t_i/1 \leq i \leq n] \approx_{SPEC} u[x_i \leftarrow t_i/1 \leq i \leq n].$$

Let $1 \leq i \leq n$. If $x_i \in var(u)$, then $sort(t_i) = sort(x_i) \in BS$ and thus $t_i \in BT$. Hence

$$t' := u[x_i \leftarrow t_i/1 \leq i \leq n] \in BT.$$

Furthermore, PAR is semantically consistent w.r.t. <BPAR,K>:
Let $A \in K$. $G_{BSPEC(A)}$ is extended to a SPEC-algebra A' by defining

$$A'_s = G_{SPEC(A),s}$$

for all $s \in S-BS$,

$$\sigma_{A'}(a_1, \ldots, a_n) = ht$$

for all $\sigma \in OP-OP'$, $<\sigma(x_1, \ldots, x_n), t> \in E-BE$, $a_i \in A'$ and $h: X \longrightarrow A'$ with $hx_i = a_i$; and

$$\sigma_{A'} = \sigma_{G_{SPEC(A)}}$$

for all $\sigma \in OP'$.
Therefore, Thms. 2.10 (ii) and 2.8 (v),(i),(ii) imply that PAR is a correct extension of <BPAR,K>. □

Two simple sufficient conditions for X-completeness and X-consistency, respectively, read as follows:

2.15 Proposition

(i) If for all $s \in BS$ $OP_s - BOP_s = \emptyset$, then PAR is X-complete w.r.t. BPAR.

(ii) If for all $s \in BS$ $E_s - BE_s = \emptyset$, then PAR is X-consistent w.r.t. BPAR.

Proof:

(i) Let $s \in BS$ and $t \in T_s$. By assumption, $t \in BT$. Hence PAR is X-complete w.r.t. BPAR.

(ii) Let $t, t' \in BT$ with $t \approx_{SPEC} t'$. By assumption, no equation of E-BE can be applied to t. Hence $t \approx_{BSPEC} t'$, and PAR is X-consistent w.r.t. BPAR. □

The third group of criteria for correct extensions concerns the case where parameter and base specification coincide, i.e. PSPEC = BSPEC. In this case correct extensionality agrees with persistency (cf. 1.20):

2.16 Proposition

Let PSPEC = BSPEC. PAR is a correct extension of <BPAR,K> iff PAR is persistent w.r.t K.

Proof:

Since PSPEC = BSPEC, we have $U_{EXT} = U_{PAR}$ and by Prop. 1.9 ("universal property" of F_{BPAR}),

$$F_{BPAR}(A) \cong A$$

for all $A \in K$. Hence persistency of PAR implies

$$U_{EXT} F_{PAR}(A) = U_{PAR} F_{PAR}(A) \cong A,$$

while correct extensionality yields

$$U_{PAR} F_{PAR}(A) = U_{EXT} F_{PAR}(A) \cong F_{BPAR}(A) \cong A. \ \square$$

Propositions 2.16 and 2.15 are combined to a useful persistency criterion:

2.17 Theorem

For all $A \in K$ and $s \in PS$ let A_s be nonempty. PAR is persistent w.r.t. K if PAR is a correct extension of <BPAR,K> and for all $s \in PS$

$$BOP_s - POP_s = BE_s - PE_s = \emptyset.$$

Proof:

By assumption and Prop. 2.15, BPAR is X-complete and X-consistent w.r.t. <PSPEC,PSPEC>. Thus by Thm. 2.10 (ii),(iii) and Thm. 2.8 (i),(ii), BPAR is a correct extension of <<PSPEC ,PSPEC>,K>. Since correct extensions are "closed under composition", PAR is a correct extension of <<PSPEC,PSPEC>,K>. Hence Prop. 2.16 implies that PAR is persistent w.r.t. K. □

2.18 Example (array)

Let BPAR = <entry,array> (cf. 1.5), PAR = <entry, array1> (cf. 2.2) and K be the class of entry-algebras defined in 1.11. Let BSIG be the signature of array. We have shown in example 1.11 that BPAR is correct w.r.t. some $BF:K \longrightarrow Alg(BSIG)$. From Example 2.2 we conclude that PAR is logically consistent w.r.t. BF (cf. 2.3). Therefore, PAR is a correct extension of <BPAR,K> by Prop. 2.6(ii). Inspection of array (1.5) immediately implies

$$sort(BOP-POP) = sort(BE-PE) = \{\underline{array}\}.$$

But $\underline{array} \notin PS$ so that by Thm. 2.17, PAR is persistent w.r.t. K. □

If PSPEC=BSPEC, then completeness and consistency simplify as follows:

2.19 Proposition

Let PSPEC = BSPEC.

(i) PAR is complete w.r.t. <BPAR,K> iff for all $A \in K$, $s \in PS$ and $t \in G(A)_s$ there is $a \in A$ with $t \equiv_{SPEC(A)} a$.

(ii) PAR is consistent w.r.t. <BPAR,K> iff for all $A \in K$ and $a,a' \in A$ $a \equiv_{SPEC(A)} a'$ implies $a = a'$.

(iii) PAR is X-complete w.r.t. BPAR iff for all $w \in S^*$, $s \in PS$, $\sigma \in OP_{w,s}$-POP and $t \in T_w$ with $op(t) \subseteq \bigcup_{s \in S-PS} OP_s$ there is $t' \in T_{POP}$ with $\sigma t \approx_{SPEC} t'$.

(iv) PAR is X-consistent w.r.t. BPAR iff for all $t,t' \in T_{POP}$ $t \approx_{SPEC} t'$ implies $t \approx_{PSPEC} t'$.

Proof:

Let $A \in K$. Assume for the moment that X = A.
Since for all $a \in A$ $id^*_A a = a$, we have $\langle t, id^*_A(t)\rangle \in \Delta(A)$ for all $t \in T_{POP}$ so that (i) holds true (cf. 1.12). Furthermore, for all $a,a' \in A$ $a \equiv_{PSPEC(A)} a'$ implies
$a = id^*_A a = id^*_A a' = a'$ because A satisfies $PE \cup \Delta(A)$.

Hence (ii) holds true. (iii) and (iv) are immediate consequences of Lemma 2.12 and Def. 2.9, respectively. □

We conclude from Prop. 2.19 (i),(ii) that in case PSPEC = BSPEC completeness and consistency agrees with sufficient completeness resp. consistency as defined in Guttag,Horning /25/,p.35 f.

2.20 Corollary

Let K be the class of all PSPEC-algebras A with $A_s \neq \emptyset$ for all $s \in PS$.

(i) If SPEC is derived from PSPEC, then PAR is persistent w.r.t. K.

(ii) If PAR is persistent w.r.t. K, then for all $w \in S^*, s \in PS, \sigma \in OP_{w,s} - POP$ and $t \in T_w$ with $op(t) \subseteq \bigcup_{s \in S-PS} OP_s$ there is $t' \in T_{POP}$ such that $\sigma t \approx_{SPEC} t'$.

Proof:

(i) follows from Prop. 2.16 and Thm. 2.14. (ii) is a consequence of Prop. 2.16, Thm. 2.10 (i) and Prop. 2.19 (iii). □

3. Overview of the criteria for completeness and consistency

We give an introduction to the completeness and consistency criteria which will be developped in detail in chapters 4 - 11. Explaining their interrelationship from the point of view of their proof is different from describing it from an application point of view. Hence we will do both, but separately. The latter will be done by the stepwise representation of decision graphs for completeness and consistency, respectively. A path from the root to a leaf corresponds to a set of sufficient conditions for completeness resp. consistency. Each leaf L is labelled by the number of the theorem which says that the set of conditions preceding L implies completeness resp. consistency. The "used"-hierarchy of these theorems will be represented by two additional graphs.

3.1 Completeness

The decision graph for completeness will be given in three parts. The first one uses two properties of a binary relation R on T the first of which is in some sense dual to the second:

- For all $\sigma, \tau \in OP$ let $\sigma >_R \tau$ if there is $\langle l,r\rangle \in R$ with $root(l) = \sigma$ and $\tau \in op(r)$. A term rewriting system R (Def. 4.4) is directly decreasing if for all $\langle l,r\rangle \in R$ and all subterms t of r

  $root(t) >_R^* root(l)$ implies $arg(l) \sqsupset_{lex} arg(t)$

  where $\sqsupset_{lex}$ is a lexicographic extension (cf. 6.1) of the subterm relation (cf. 4.8).

Directly decreasing relations are given in 6.9, 6.12, 7.9, 8.6 and 8.7.

- R is base-total w.r.t. $BOP' \subseteq BOP$ if for all $w \in BS^*, \sigma \in OP-BOP$ with arity w and $u \in T_{BOP',w}$ with $sort(op(u)) \cap PS = \emptyset$ and $sort(var(u)) \subseteq PS$ there are $t \in T_{BOP',w}$, $\langle \sigma(t),r\rangle \in R$ and

$f \in Z(T)$ with $ft = u$.

Base-total relations are given in 7.8, 7.9, 8.6 and 8.7.

Intuitively, R is directly decreasing if the arguments of recursive function calls in R-reductions decrease.

By 6.10, 6.5 and 5.4, "directly decreasing" implies "normalizing", i.e. all R-reductions are terminating. The definition of "directly decreasing" in 6.8 is a little more general. On the other hand, "R base-total" means that each $\xrightarrow[R]{}$-normal term (cf. 4.1, 4.3) whose sort belongs to BS is a BOP-term. "Directly decreasing" and "base-total" correspond to the terms "weight-decreasing" resp. "generating" used in Ehrig, Kreowski, Padawitz /19/ in the following way:

On one hand "R directly decreasing" is more general than "R weight-decreasing" since it allows in $\langle l,r \rangle \in R$ more than one operation symbol on the right-hand side with the "degree" of root(l) (even "nested recursion"). On the other hand, adapting "directly decreasing" such that it will follow from "weight-decreasing" would require the condition

$$\text{size}(\text{arg}(l)) >_{\text{lex}} \text{size}(\text{arg}(t)) \qquad (*)$$

instead of

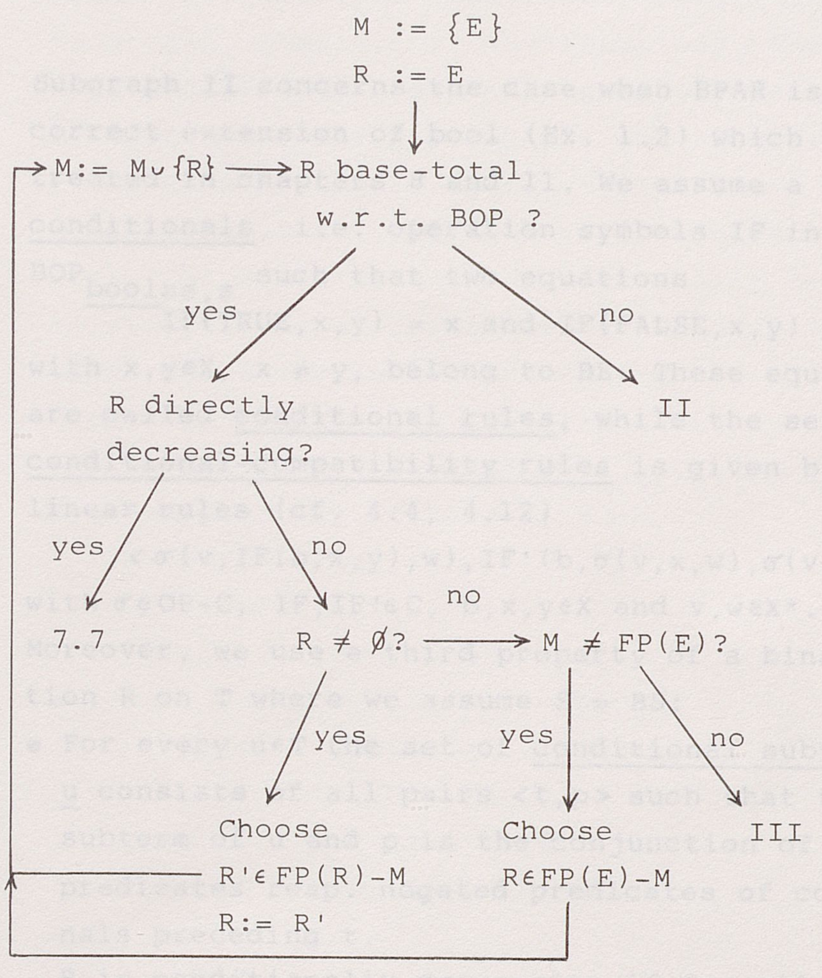
$$\text{arg}(l) \supset_{\text{lex}} \text{arg}(t)$$

in the definition of "directly decreasing". But then we would loose the property that directly decreasing relations are normalizing. We have not seen examples where (*) is needed to show completeness; but normalization is the crucial property of directly decreasing relations when they are used in consistency proofs (cf. 3.2).

"Base-total" implies "generating", while "base-total" is decidable and "generating" is not because it involves the congruence of the base specification.

The first part of the decision graph looks as follows:

I



We first ask whether R = E is base-total w.r.t BOP and directly decreasing. If so, then PAR is complete w.r.t. <BPAR,K> by Thm. 7.7. Otherwise we choose a proper subset of R for a new R and again check the two properties. If one of these R's is not base-total, we go to the subgraph denoted by II and described below. If R is empty, another subset of E has to be chosen for R. If the check for base-totality and direct decrease has failed for all subsets of E (M = FP(E)), we arrive at subgraph III. The reason for the succession of tests in graph I is the "trade-off" between base-totality and direct decrease: If R⊆E is base-total, then any <u>superset</u> R'⊆E of R is base-total, too, but not vice versa.

If R is directly decreasing, then any <u>subset</u> of R is directly decreasing, too, but again not vice versa.

Subgraph II concerns the case when BPAR is a correct extension of bool (Ex. 1.2) which will be treated in chapters 8 and 11. We assume a set C of <u>conditionals</u>, i.e. operation symbols IF in $\bigcup_{s \in BS} BOP_{bool\,s\,s,s}$ such that two equations

$$IF(TRUE,x,y) = x \text{ and } IF(FALSE,x,y) = y$$

with $x,y \in X$, $x \neq y$, belong to BE. These equations are called <u>conditional rules</u>, while the set CCR of <u>conditional-compatibility rules</u> is given by all linear rules (cf. 4.4, 4.12)

$$<\sigma(v,IF(b,x,y),w),IF'(b,\sigma(v,x,w),\sigma(v,y,w))>$$

with $\sigma \in OP-C$, $IF,IF' \in C$, $b,x,y \in X$ and $v,w \in X^*$.

Moreover, we use a third property of a binary relation R on T where we assume S = BS:

- For every $u \in T$ the set of <u>conditional subterms of u</u> consists of all pairs $<t,p>$ such that t is a subterm of u and p is the conjunction of all predicates resp. negated predicates of conditionals preceding t.

  R is <u>conditionally decreasing</u> if for all $A \in K$ and $\sigma \in OP-BOP$ there is a "weight" function

  $$w_{\sigma,A}: BG(A)_{arity(\sigma)} \longrightarrow \mathbb{N}$$

  such that for all $<l,r> \in R$

  (i) l and all subterms t of r with $root(t) \in OP-BOP$ are <u>simple</u> terms, i.e. $root(t) \in OP-BOP$ and $arg(t) \in BT^*$,

  and for all $<l,r> \in R$, $f \in BZ(G(A))$ and all conditional subterms $<t,p>$ of r with simple t

  (ii) $fp \equiv_{BSPEC(A)} TRUE$ implies
  $$w_{root(t),A}(f \circ arg(t)) < w_{root(l),A}(f \circ arg(l)).$$

Roughly spoken, R is conditionally decreasing if for all $p \in BT_{\underline{bool}}$ and $f \in BZ(G(A))$

$$fp \equiv_{BSPEC(A)} TRUE \text{ and } t \xrightarrow[R]{} t'$$

imply $w_A(ft) > w_A(ft')$
for some "weight" function $w_A : BG(A)^* \longrightarrow \mathbb{N}$.
Hence, R-reductions need to be normalizing only for "logically consistent" substitutions. The idea behind conditionally decreasing relations evolves from the definition principle for recursive functions used in Boyer, Moore /7/, p. 44.
Examples of such relations will be given in 8.14 - 8.21. The definition of "conditionally decreasing" is slightly more complicated if we drop the assumption that S equals BS (cf. chapter 8).
Moreover, $f \circ arg(t)$ in condition (ii) can be replaced by any $t' \in BT^*$ with $t' \equiv_{BSPEC(A)} f \circ arg(t)$.
For example, let PSPEC = $\emptyset$, $m \in \mathbb{N}$,

BSPEC = nat +
sorts: $\underline{nat}_m$
opns: CODE: $\underline{nat} \longrightarrow \underline{nat}_m$
<: $\underline{nat}\ \underline{nat} \longrightarrow \underline{bool}$
eqns: $CODE(S^m x) = CODE(x)$ $\underline{nl3}$
$x<0 = FALSE$ $\underline{nl4}$
$0<Sx = TRUE$ $\underline{nl5}$
$Sx<Sy = x<y$ $\underline{nl6}$

and

SPEC = BSPEC +
opns: EQ: $\underline{nat}_m\ \underline{nat}_m \longrightarrow \underline{bool}$
eqns: $EQ(CODE(x), CODE(y)) =$
$= IFN(x<S^m 0 \wedge y<S^m 0,$
$EQN(x,y),$
$EQ(CODE(x), CODE(y)))$ $\underline{nl7}$

(cf. 1.3). <u>R = {nl7} is conditionally decreasing:</u>
(i) is easily checked. The only conditional subterm $\langle t,p \rangle$ of the right side of $\underline{nl7}$ with simple t is given by $t = EQ(CODE(x), CODE(y))$ and $p = \neg(x<S^m 0 \wedge y<S^m 0)$. There is an obvious initial BSPEC-algebra B with $B_{\underline{bool}} = \{true, false\}$, $B_{\underline{nat}} = \mathbb{N}$ and $B_{\underline{nat}_m} = \{0,..., m-1\}$.
Let $w_{EQ}: BG_{\underline{nat}_m} \longrightarrow \mathbb{N}$ be

defined by

$$w_{EQ}(\text{CODE}(u), \text{CODE}(u')) = u_B + u'_B$$

Let $f \in BZ(G)$ and $fp \equiv_{BSPEC} \text{TRUE}$, i.e. $(fp)_B = \text{true}$.
The generalized version of condition (ii) is satisfied if there is $t' \in BT^*$ with $t' \equiv_{BSPEC} f \circ \arg(t)$ and $w_{EQ}(t') < w_{EQ}(f \arg(1))$.
Now $(fp)_B = \text{true}$ w.l.o.g. implies $(fx)_B \geq m$ and thus $(fx)_B = m + n = (S^{m+n}O)_B$ for some $n \in \mathbb{N}$.

So let $\quad t' = \langle \text{CODE}(S^n O), \text{CODE}(fy) \rangle$.

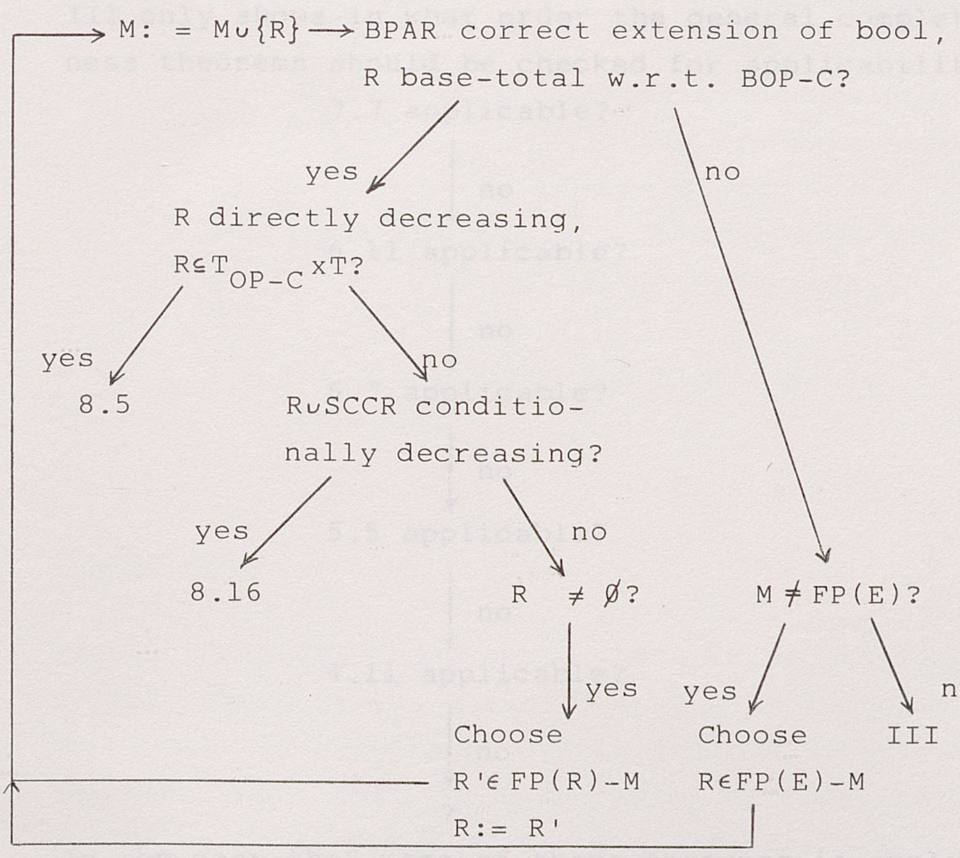
Then $\quad t'_1 \equiv_{BSPEC} \text{CODE}(S^{m+n} 0) \equiv_{BSPEC} \text{CODE}(fx)$

so that $\quad t' \equiv_{BSPEC} f \circ \arg(t)$, and

$$w_{EQ}(t') = n + (fy)_B < m + n + (fy)_B = (fx)_B + (fy)_B = w_{EQ}(f \circ \arg(1)).$$

A more complex example using the generalized version of condition (ii) is given by 8.13 - 8.15.
Let SCCR be the set of all rules of CCR with simple lefthand side. Then subgraph II has the following shape:

II



Since base-totality w.r.t. BOP implies base-totality w.r.t BOP-C, we are now more general than in the first part of our decision graph, provided that BPAR extends bool. Moreover, this assumption is crucial for the conditional decrease of R∪SCCR which we may test if the test on direct decrease has failed. For the explanation of the cycles see the remarks following graph I.

The completeness conditions of subgraph I are decidable while those of subgraph II are decidable with respect to BPAR, i.e. there are algorithms to check them provided that for all $A \in K \equiv_{BSPEC(A)}$ is decidable. In contrast to that subgraph III contains criteria which are too general to be decidable but which reflect the proof-theoretical steps from the definition of completeness to sufficient conditions that can be verified algorithmically. Since those criteria use terms whose definition would exceed the scope of this overview, subgraph III only shows in what order the general completeness theorems should be checked for applicability.

III

$$
\begin{array}{c}
\text{7.7 applicable?} \\
\downarrow \text{ no} \\
\text{6.11 applicable?} \\
\downarrow \text{ no} \\
\text{6.7 applicable?} \\
\downarrow \text{ no} \\
\text{5.5 applicable?} \\
\downarrow \text{ no} \\
\text{4.11 applicable?} \\
\downarrow \text{ no} \\
?
\end{array}
$$

In the case that none of these theorems is applicable but PAR is consistent w.r.t.<BPAR,K>, completeness may follow from semantical completeness (cf. 2.7).

That the criteria presented in chapters 4-8 are sufficient for completeness will not be proved seperately for each set of conditions. Instead, the main theorems that concern completeness constitute the following hierarchy where the source of each arrow uses its target.

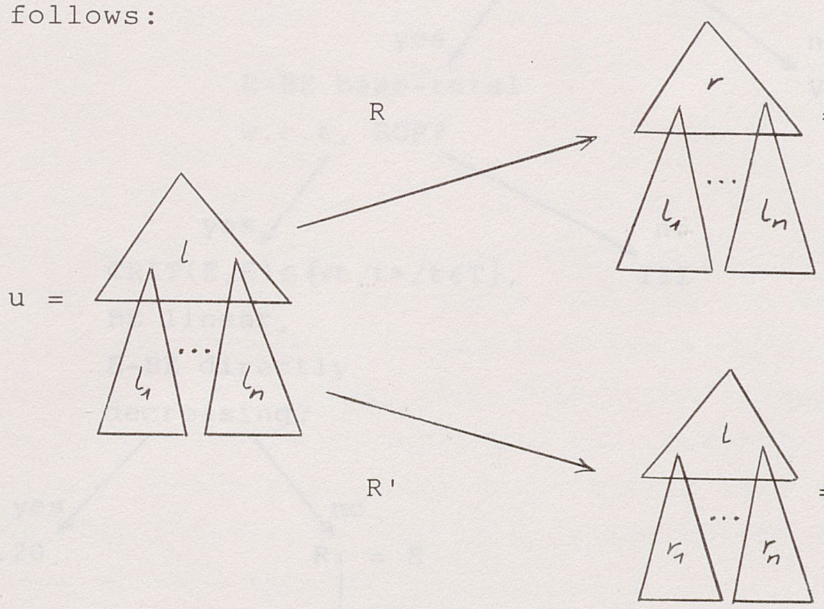
$$\begin{tikzcd}
 & & & 8.5 \arrow[d] & \\
 & & & 7.7 \arrow[dl] \arrow[dr] & \\
 & & 6.11 \arrow[dl] \arrow[d] & & 7.2 \text{ (Base-total relations are base-complete)} \\
 & 6.7 \arrow[dl] \arrow[d] & 6.10 \text{ (Directly decreasing relations are recursively decreasing)} & & \\
 5.5 \arrow[d] \arrow[dr] & 6.5 \text{ (Recursive path orderings are simplification orderings)} & & & \\
 4.11 & 5.3 \text{ (Simplification orderings are well-founded)} & & &
\end{tikzcd}$$

3.2 Consistency

The decision graph for consistency is separated into five parts. Parts I and II deal with the case where E-BE is base-total, parts III and IV treat the case where BPAR is an extension of bool and - analogous to subgraph II for completeness - E-BE might be base-total w.r.t. BOP minus some set C of conditionals only. Subgraph V summarizes the consistency theorems with weaker but undecidable criteria.

We need the notion of a critical pair of terms used in the Knuth-Bendix (semi-) algorithm for transforming a normalizing term rewriting system into a uniquely normalizing one (cf. Knuth, Bendix /43/ and Huet /31/):

- Given two term rewriting systems R and R', $\langle l,r\rangle \in R$ and $\langle l_1,r_1\rangle,\ldots,\langle l_n,r_n\rangle \in R'$, $\langle t,t'\rangle \in T^2$ is called a critical pair of $\langle R,R'\rangle$ if $l_1,\ldots,l_n$ overlap l in operation symbols and t,t' result from replacing l by r and $l_1,\ldots,l_n$ by $r_1,\ldots,r_n$, respectively. The situation looks roughly as follows:



$\langle t,t'\rangle$ is called including if there is a "proper path" from a "leaf" of some l_1 to a "leaf" of l in the term named u in the diagram.

The papers referred to above only use simple critical pairs, i.e. critical pairs where n equals 1. CRIT(R,R') and SCRIT(R,R') denote the sets of critical resp. simple critical pairs of $\langle R,R'\rangle$.

The formal definition of a critical pair is given in 9.3. An example of a non-simple critical pair is presented in 9.6.

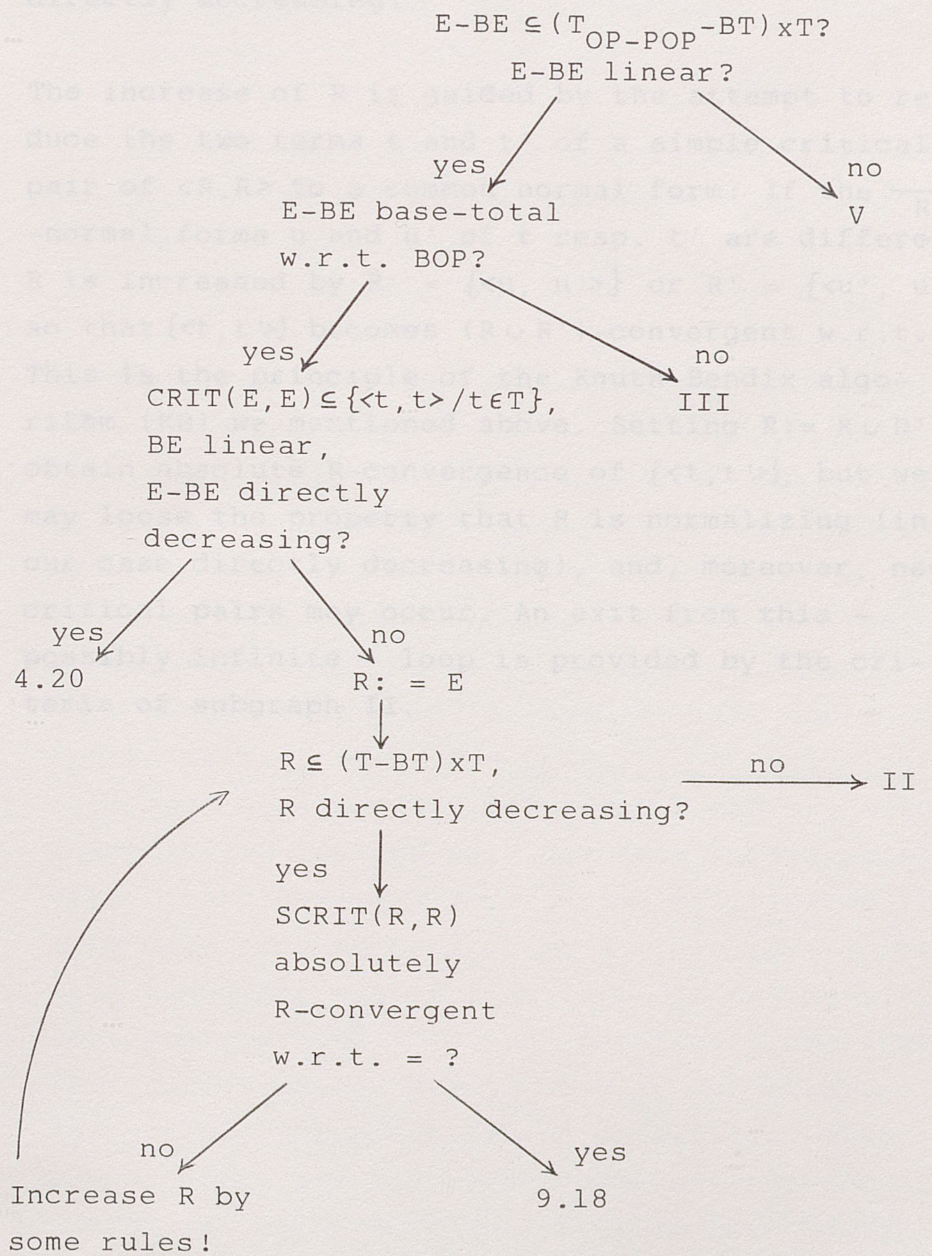
- Given relations R and R' on T, $M \subseteq T^2$ is called <u>absolutely R-convergent w.r.t. R'</u> if for all $\langle t,t'\rangle \in M$ there is $\langle u,u'\rangle \in R'$ such that t and t' are R-reducible to u resp. u' (cf. 4.1).

In Example 9.19 we show the absolute R-convergence of SCRIT(R,R) w.r.t. equality where R is a set of equations in the theory of int (cf. 7.9).

- $R \subseteq T^2$ is <u>linear</u> if for all $\langle l,r\rangle \in R$ l has unique variable occurrences.

Now we are ready to present the first part of our decision graph for consistency:

I



Here we have two sets of conditions each of which is sufficient for consistency of PAR w.r.t.<BPAR,K>. The main property in the first set is the mutual independence of E-reductions: There are only trivial critical pairs of $<E,E>$. If this does not hold, we consider the case where E is directly decreasing and ask for absolute E-convergence of SCRIT(E,E) w.r.t. equality. If the answer is negative, the relation R is increased by additional equations such that - eventually - SCRIT(R,R) is absolutely R-convergent w.r.t. = and R is still directly decreasing.

The increase of R is guided by the attempt to reduce the two terms t and t' of a simple critical pair of $<R,R>$ to a common normal form: If the $\xrightarrow[R]{}$-normal forms u and u' of t resp. t' are different, R is increased by $R' = \{<u, u'>\}$ or $R' = \{<u', u>\}$ so that $\{<t,t'>\}$ becomes $(R \cup R')$-convergent w.r.t. =. This is the principle of the Knuth-Bendix algorithm (KB) we mentioned above. Setting $R := R \cup R'$ we obtain absolute R-convergence of $\{<t,t'>\}$, but we may loose the property that R is normalizing (in our case directly decreasing), and, moreover, new critical pairs may occur. An exit from this - possibly infinite - loop is provided by the criteria of subgraph II.

Before turning to subgraph II we want to make some remarks on the paper "Proofs by Induction in Equational Theories with Constructors" (see Huet, Hullot /32/) because it also treats the problem of deciding identities in initial algebras with the help of KB:
Huet and Hullot extend KB to the "inductive completion algorithm" (HH) which, given a specification SPEC = <S,OP,E> that satisfies the "principle of definition", has the following properties:

(1) If HH applied to some OP-equation e stops with "success", then e holds true in G_{SPEC}.

(2) Assumed that HH does not stop with "failure", HH decides the complement of $\Delta(G_{SPEC})$, the equational diagram of G_{SPEC}, i.e. the set of all OP-equations that hold true in G_{SPEC}.

Note that HH does not decide $\Delta(G_{SPEC})$. Nourani /51/ has shown that not even a semidecision procedure for $\Delta(G_{SPEC})$ exists. However HH can be used for consistency proofs, and Huet/Hullot have given examples where it works quite well. But the "principle of definition" imposes a strong limitation on this method which makes it necessary to apply further consistency proof strategies like those presented in this work. The <u>principle of definition</u> says that any two SPEC-congruent terms are SPEC-congruent to one and only one term formed solely from "constructors". This is too restrictive if SPEC uses constructors which are already specified by a non-empty set of equations: If we identify the constructors with BOP, the operation symbols of our base specification BSPEC, most of our examples do not satisfy the principle of definition. Often the constructors are not even "commutative-associative operators" which Huet/Hullot claim that HH can be extended to (Huet, Hullot /32/, p. 249). This limitation of Huet/Hullot's approach has also been recognized by Bidoit /6/ but his proposal to circumvent the principle of definition seems to be nothing else than the classical method of proving equations in G_{SPEC} by structural induction and rewriting.

Prior to Huet and Hullot, Musser /48/ and Goguen /23/ have constructed two algorithms which are more closely bound to KB and which also obey properties (1) and (2). The principle of both is the same, so let us confine to one and call it MG. The main difference between MG and HH is that MG does not require the principle of definition but the fact that SPEC is an extension of bool and contains for each $s \in S$ a specification of the equality on $G_{SPEC,s}$. Comparing this requirement to the principle of definition we first note that many data type specifications which do not satisfy the latter allow equality axiomatizations. On the other hand, if SPEC fulfils the principle of definition, then we obtain equality specifications:

Let $SPEC' = SPEC \cup bool \cup \bigcup_{s \in S} SPEC_s$ where

$SPEC_s$

opns: $EQ_s : ss \longrightarrow \underline{bool}$

eqns: $EQ_s(\sigma,\sigma) = TRUE$ for all $\sigma \in BOP_s$

$EQ_s(\sigma(x_1,..,x_n),\sigma(y_1,..,y_n))$

$= EQ_{s1}(x_1,y_1) \wedge ... \wedge EQ_{sn}(x_n,y_n)$

for all $\sigma \in BOP_{s1...sn,s}$

$EQ_s(\sigma(x_1,...,x_n),\tau(y_1,...,y_m))$

for all $\sigma,\tau \in BOP$ with $\sigma \neq \tau$

A simple induction on size(t) resp. size(t) + size(t') yields for all $s \in S$ and $t, t' \in BG_s$ with $t \neq t'$

$$EQ_s(t,t) \equiv_{SPEC'} TRUE$$

and

$$EQ_s(t,t') \equiv_{SPEC'} FALSE.$$

Hence, by the principle of definition, for all $s \in S$ and $t,t' \in G_s$

$$t \equiv_{SPEC} t' \text{ implies } EQ_s(t,t') \equiv_{SPEC'} TRUE$$

and

$$t \not\equiv_{SPEC} t' \text{ implies } EQ_s(t,t') \equiv_{SPEC'} FALSE$$

Thus $SPEC_s$ specifies the equality on $G_{SPEC,s} = G_{SPEC',s}$.

Since all ground terms of sort <u>bool</u> are TRUE, FALSE or have the form $EQ_s(u,u')$ for some $s \in S$ and $u,u' \in G_s$, SPEC' is an extension of bool by Lemma 8.1.

Therefore, MG puts weaker conditions on SPEC than HH. But we admit that HH tackles the validity problem for equations more directly. In order to see this and to give more insight into the principles of HH and MG, let us sketch both algorithms and their correctness with respect to (1) and (2): Let e be the equation to be proved for validity in G_{SPEC}, and $SPEC(e) = \langle S, OP, E \cup \{e\} \rangle$.

MG is based on the easily provable fact that under the assumption for MG (existence of equality specifications, see above) e holds true in G_{SPEC} if SPEC(e) is logically consistent, i.e. TRUE and FALSE are not SPEC(e)-congruent. Now MG initializes two term variables t and t' with TRUE resp. FALSE, starts KB on $R = E \cup \{e\}$ and, in every loop of KB, sets t and t' to the current $\xrightarrow[R]{}$-normal forms of t resp. t'. MG stops with "failure" if R is no more normalizing. MG stops with "success" if R is normalizing and confluent and t,t' have different $\xrightarrow[R]{}$-normal forms. MG stops with "disproof" if t and t' have equal $\xrightarrow[R]{}$-normal forms.

If MG stops with "success", then R is normalizing and confluent, and t,t' have different $\xrightarrow[R]{}$-normal forms. Thus $e \in \Delta(G_{SPEC})$. Otherwise SPEC(e) would be logically inconsistent so that $t \equiv_{SPEC(e)} t'$. Since R is confluent, the $\xrightarrow[R]{}$-normal forms of t,t' would agree.

Now assume that MG neither stops with "failure" nor with "disproof". Let R_∞ be the - possible infinite - relation generated by MG. SCRIT(R_∞,R_∞) is absolutely R_∞-convergent w.r.t. = because for all $\langle u,u'\rangle \in$ SCRIT(R_∞,R_∞) there is an iteration of MG where $\langle u,u'\rangle$ is a simple critical pair of $\langle R,R\rangle$, and $\{\langle u,u'\rangle\}$ becomes absolutely R-convergent in the next iteration. Therefore, by Thm. 9.15 or Huet /30/, Thm. 3.2, R_∞ is confluent. Since R_∞ is normalizing, there will be some iteration of MG where t,t' are $\xrightarrow[R]{}$-normal. Since MG does not stop with "disproof", t $\neq$ t'. Thus TRUE $\not\equiv_{SPEC(e)}$ FALSE because R_∞ is confluent. Hence $e \in \Delta(G_{SPEC})$.
If MG stops with "disproof", there is an iteration of MG where the $\xrightarrow[R]{}$-normal forms of t,t' agree. Thus TRUE $\equiv_{SPEC(e)}$ FALSE so that $(e \in \Delta(G_{SPEC})$.

Therefore, (1) and (2) hold true for MG instead of HH.
Let us now turn to HH. In addition to the principle of definition we assume that for all $\langle l,r\rangle \in E\cup\{e\}$ $l \notin BT$. HH starts KB on R = E$\cup$\{e\}, stops with "failure" if R is no more normalizing and stops with "success" if R is normalizing and confluent. Let $\langle u,u'\rangle$ and $\langle u',u\rangle$ be the candidates to increase R in some loop of KB. HH chooses that one where the first component does not belong to BT$\cup$X. If u,u'$\in$BT$\cup$X, then HH stops with "disproof".
Let e = $\langle l,r\rangle$ and, for an OP-term t, let $\bar{t}$ denote the unique BOP-term with $t \equiv_{SPEC} \bar{t}$ given by the principle of definition.
If HH stops with "success", then $e \in \Delta(G_{SPEC})$. Otherwise there would be some $f \in Z(G)$ with $fl \not\equiv_{SPEC} fr$ so that $\overline{fl} \neq \overline{fr}$. On the other hand, we have

$$\overline{fl} \equiv_{SPEC} fl \equiv_{SPEC(e)} fr \equiv_{SPEC} \overline{fr}.$$

Since R is normalizing and confluent, the

$\xrightarrow[R]{}$-normal forms u, u' of $\overline{fl}$ resp. $\overline{fr}$ would agree. Since for all $\langle l,r\rangle \in R$ $l \notin BT \cup X$, we would obtain $\overline{fl} = u = u' = \overline{fr}$ which contradicts $\overline{fl} \neq \overline{fr}$ derived above.

Assume that HH neither stops with "failure" nor with "disproof". Then the relation R_∞ generated by HH is confluent. This can be shown for HH as for MG (see above). Since for all $f \in Z(G)$

$$\overline{fl} \equiv_{SPEC} fl \equiv_{SPEC(e)} fr \equiv_{SPEC} \overline{fr},$$

the $\xrightarrow[R_\infty]{}$-normal forms u_f, u'_f of $\overline{fl}$ resp. $\overline{fr}$ agree, and we get $\overline{fl} = u_f = u'_f = \overline{fr}$ because for all $\langle l,r\rangle \in R_\infty$ $l \notin BT \cup X$. Thus for all $f \in Z(G)$ $fl \equiv_{SPEC} fr$, i.e. $e \in \Delta(G_{SPEC})$.

If HH stops with "disproof", there are $t, t' \in BG$ with $t \neq t'$, but $t \equiv_{SPEC(e)} t'$. Hence, by the principle of definition, SPEC-congruence must be different from SPEC(e)-congruence so that $e \notin \Delta(G_{SPEC})$.

Therefore, HH satisfies (1) and (2).

The reader who is familiar with Huet, Hullot /32/ will notice that we have presented a simplified version of HH. But our aim here was to give the "skeletons" of KB, MG and HH and to suppress all details which make it difficult to perceive that the algorithms are correct with respect to (1) and (2).

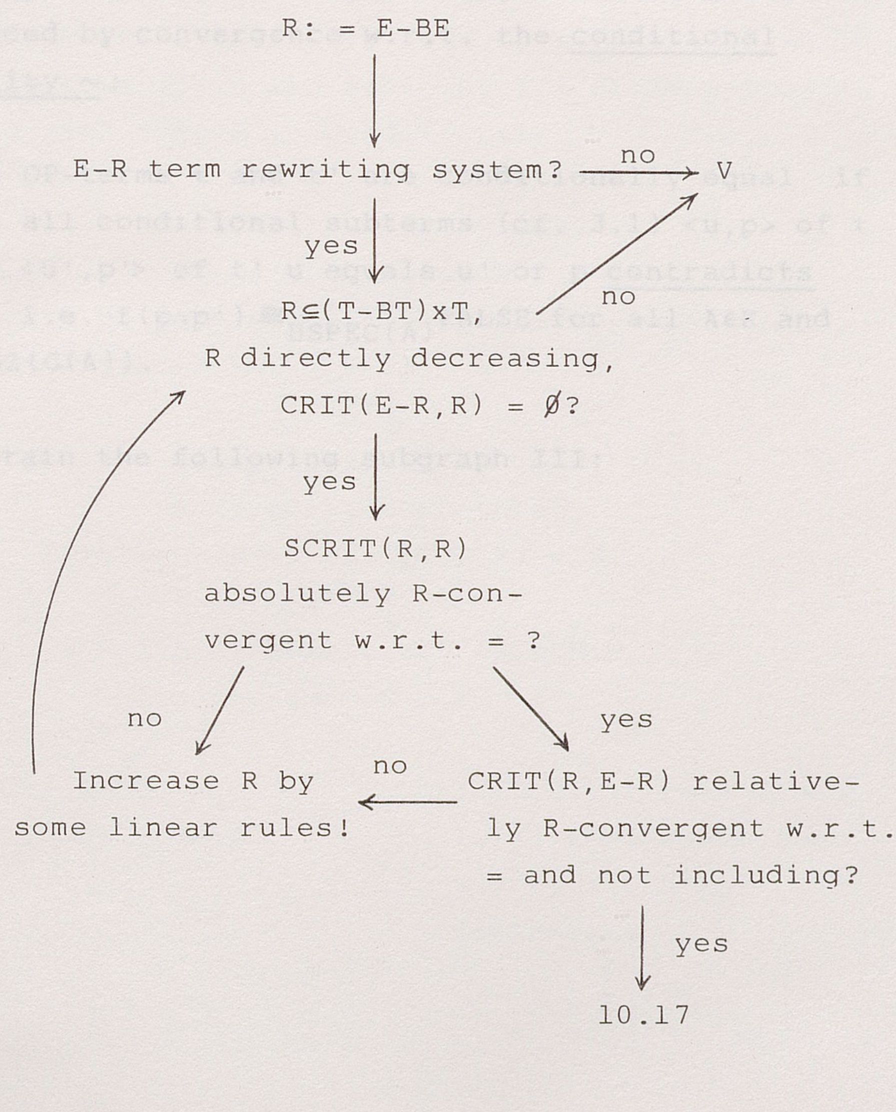
Subgraph II uses the following confluence property which considers reductions via E-R:

- Given relations R, R' on T, $M \subseteq T^2$ is called <u>relatively R-convergent w.r.t. R'</u> if for all $\langle t,t'\rangle \in M$ there is $\langle u,u'\rangle \in R'$ such that $t \xrightarrow[R]{*} \overset{\Delta}{\underset{E-R}{\Longrightarrow}} \xrightarrow[R]{*} u$ and $t' \xrightarrow[R]{*} u'$. $\underset{E-R}{\Longrightarrow}$ denotes the parallel (E-R)-reduction relation (cf. 10.5) and $\overset{\Delta}{\underset{E-R}{\Longrightarrow}}$ its reflexive closure.

We introduce the notions "relative R-convergence" and "relative confluence" (cf. 10.6 or Padawitz /53/, Def. 3.1: "SPEC commutes with recursive E0-

reductions") as an alternative to "confluence modulo the equivalence closure of $\xrightarrow[E-R]{}$" used in Peterson, Stickel /55/ and Huet /30/. Considering $(\xrightarrow[E-R]{})^{-1}$ in this confluence property implies that it depends on convergence of critical pairs which come from overlappings between lefthand sides of R and righthand sides of E-R. In 10.1 - 10.4 we show in detail why Huet's "confluence modulo" is not appropriate if E-R is "non-permutative", e.g. if BSPEC = array. Jeanrond /35/ came to the same conclusion with respect to Peterson/Stickel's "confluence modulo" when he failed to prove that property for set (cf. 8.7). This limitation also applies to Jouannaud /37/ who generalizes Peterson/ Stickel /55/ and Huet /30/.

II



In the lower part of subgraph I we have searched for a directly decreasing relation R such that R includes E and SCRIT(R,R) is absolutely R-convergent w.r.t. =. Now we do the same except that R must include only E-BE. But we have to check in addition that CRIT(E-R,R) is empty and CRIT(R,E-R) is relatively R-convergent w.r.t. =. Escaping from the loop we will arrive at subgraph V.
If E-BE is not base-total w.r.t. BOP (cf. subgraph I), we continue with subgraph III where we first ask whether BPAR is an extension of bool. In that case we proceed similarly to the lower part of subgraph I, now considering the conditionals C, the conditional rules CR and the conditional-compatibility rules CCR (see the remarks to completeness subgraph II). Therefore, convergence w.r.t. = is replaced by convergence w.r.t. the <u>conditional</u> <u>equality</u> $\sim$:

- Two OP-terms t and t' are conditionally equal if for all conditional subterms (cf. 3.1) <u,p> of t and <u',p'> of t' u equals u' or p <u>contradicts</u> p', i.e. $f(p \wedge p') \equiv_{BSPEC(A)} FALSE$ for all $A \in K$ and $f \in BZ(G(A))$.

We obtain the following subgraph III:

III

BPAR correct extension of bool,
E-BE base-total w.r.t. BOP-C?
yes ↙ no ↘ V
R: = E-(CR∪CCR)
↓
R⊆(T_{OP-C}-BT)xT,
R directly decreasing, $\xrightarrow{\text{no}}$ IV
CRIT(CCR,R) = ∅?
yes ↓
SCRIT(R,R)
absolutely R-convergent
w.r.t ∼ ?
no ↙ yes ↘
Increase R by
some rules! 11.10

We search for a directly decreasing relation R such that R includes E-(CR∪CCR) and SCRIT(R,R) is absolutely R-convergent w.r.t. conditional equality. An exit from the loop is provided by subgraph IV which differs from II in the same way III differs from I, namely with regard to conditionals.

IV

R: = E-(BE∪CR∪CCR)
R': = R∪CR∪CCR
↓
E-R' term rewriting system, $\xrightarrow{\text{no}}$ V
CRIT(CCR,E-R') = ∅?
yes ↓ no ↗ V
R⊆(T_{OP-C}-BT)xT,
R directly decreasing,
CRIT(E-R',R) = CRIT(CCR,R) = ∅?
yes ↓
SCRIT(R,R) absolutely (R∪CCR)-
convergent w.r.t. ∼ ?
no ↙ yes ↘
Increase R by
some linear rules! $\xleftarrow{\text{no}}$ CRIT(R,E-R') relatively (R∪CCR)-convergent
w.r.t. ∼ and not
including?
↓ yes
11.11

In Examples 11.15 and 11.16 we show the relative $(R \cup CCR)$-convergence of CRIT(R,E-R) w.r.t. $\sim$ where E is given by the equations of arrayl resp. setl and $R = \{\underline{a5}, \underline{a6}\}$ resp. $R = \{\underline{s5}, \underline{s6}\}$ (cf. 2.2 resp. 8.7).

Finally, subgraph V summarizes the more general consistency theorems which reflect steps in the proof that the criteria of subgraphs I - IV are sufficient for consistency. The conditions of I and II are decidable, those of III and IV can be checked by algorithms if for all $A \in K$ $\equiv_{BSPEC(A)}$ is decidable; that the assumptions of the theorems in V are valid is in general not computable. (Cf. the corresponding remarks on the decidability of completeness conditions in 3.1.)

V

9.16 applicable?

$\downarrow$ no

4.19 applicable?

$\downarrow$ no

10.15 applicable?

$\downarrow$ no

10.7 applicable?

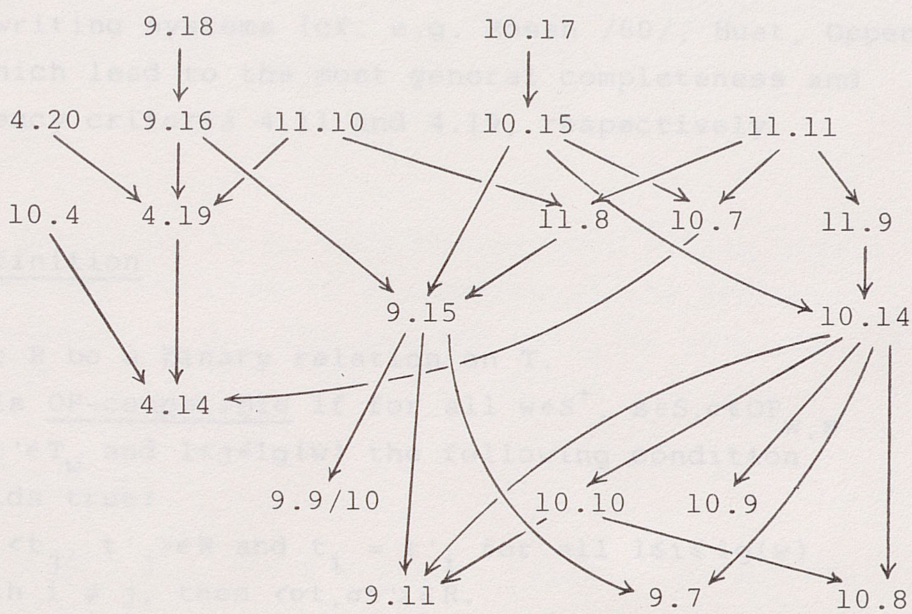
$\downarrow$ no

10.4 applicable?

$\downarrow$ no

?

The following graph represents the "used"-hierarchy of the main theorems and lemmas that concern consistency.



4. Term reductions

This chapter introduces basic notions of the theory of term rewriting systems (cf. e.g. Rosen /60/, Huet, Oppen /34/) which lead to the most general completeness and consistency criteria 4.11 and 4.19, respectively.

4.1 Definition

Let R be a binary relation on T.

R is OP-compatible if for all $w \in S^+$, $s \in S$, $\sigma \in OP_{w,s}$, $t, t' \in T_w$ and $1 \leq j \leq lg(w)$ the following condition holds true:

If $\langle t_j, t'_j \rangle \in R$ and $t_i = t'_i$ for all $1 \leq i \leq lg(w)$ with $i \neq j$, then $\langle \sigma t, \sigma t' \rangle \in R$.

R is OP-stable if for all $\langle t, t' \rangle \in R$ and $f \in Z(T)$ $\langle ft, ft' \rangle \in R$.

R has a natural extension to Z(T): For all $f, g \in Z(T)$ $\langle f, g \rangle$ belongs to R if there is $x \in X$ with $\langle fx, gx \rangle \in R$ and for all $y \in X$ $\langle fy, gy \rangle \in R^{\Delta}$.

$\xrightarrow[R]{}$ denotes the R-reduction relation on T, i.e. the least OP-compatible and OP-stable relation on T that includes R. We write $t \xrightarrow[R]{} t'$ instead of $\langle t, t' \rangle \in \xrightarrow[R]{}$.

$\xrightarrow[R]{\Delta}$, $\xleftrightarrow[R]{}$, $\xrightarrow[R]{*}$ and $\xleftrightarrow[R]{*}$ denote the reflexive, symmetric, reflexive-transitive resp. equivalence closure of $\xrightarrow[R]{}$. If $t \xrightarrow[R]{*} t'$, then t is R-reducible to t'.

Let $A \in K$. Replacing OP, T and Z(T) by OP(A), T(A) and Z(T(A)), respectively, we obtain the notions OP(A)-compatible and OP(A)-stable, the extension of R to Z(T(A)) and the R-reduction relation on T(A).

4.2 Proposition

Let $A\in K$. $\overset{*}{\underset{E}{\longleftrightarrow}}$ resp. $\overset{*}{\underset{E(A)}{\longleftrightarrow}}$ agrees with $\approx_{SPEC}$ resp. $\approx_{SPEC(A)}$ (cf. 1.14). Moreover, $\overset{*}{\underset{E(A)}{\longleftrightarrow}}\cap G(A)$ coincides with $\equiv_{SPEC(A)}$ (cf. 1.7).

Proof:

The prosition is an easy consequence of the definitions and the general assumption at the beginning of chapter 2 that for all $s\in S$ $G(A)_s$ is nonempty. □

4.3 Definition

Let M be a set and R a binary relation on M. A sequence $(a_i)_{i\in\mathbb{N}}$ of elements of M with $\langle a_i, a_{i+1}\rangle\in R$ for all $i\in\mathbb{N}$ is an (infinite) chain of R. $a\in M$ is R-normal if for all $\langle a,b\rangle\in R$ a is equal to b. R is well-founded if chains of R do not exist.

4.4 Definition

Let $A\in K$. $\langle l,r\rangle\in T(A)^2$ is a rule if $l\notin X$ and $var(r)\subseteq var(l)$. A set of rules is called a term rewriting system.
A binary relation R on T(A) is normalizing if the R-reduction relation on T(A) is well-founded. NF(R) denotes the set of $\underset{R}{\longrightarrow}$-normal OP(A)-terms.
$\langle R,A\rangle$ is base-complete if for all $s\in BS$

$$G(A)_s\cap NF(R)\subseteq BG(A).$$

$R\subseteq T^2$ is base-complete if $\langle R,A\rangle$ is base-complete for all $A\in K$.

If S is a singleton and $G_{SPEC(A)}$ has at least two elements, then every normalizing subset R of E is a term rewriting system: If there would be $\langle x,r\rangle\in R$ with

$x \in X$, then r must contain a variable y, otherwise all ground OP(A)-terms would be SPEC(A)-congruent to r, i.e. $/G_{SPEC(A)}/ = 1$. Choosing $f \in Z(T)$ with $fy = r$, we would obtain a chain of $\xrightarrow[R]{}$, namely

$$x \xrightarrow[R]{} r \xrightarrow[R]{} fr \xrightarrow[R]{} ffr \xrightarrow[R]{} \dots$$

If there would be $\langle l,r \rangle \in R$ and $x \in X$ with $x \in var(r) - var(l)$, then

$$l \xrightarrow[R]{} fr \xrightarrow[R]{} fgr \xrightarrow[R]{} fggr \xrightarrow[R]{} \dots$$

would be chain of $\xrightarrow[R]{}$ where $f,g \in Z(T)$ satisfy $fx = l$ and $gx = r$.

Hence normalizing term relations are in general term rewriting systems. The converse is not valid. Huet, Lankford /33/ have even shown that the normalization property is undecidable. In chapters 5 and 6 we shall study sufficient conditions for this property which are weakly enough for completeness and consistency proofs of equational specifications.

4.5 Definition

Let R be a binary relation on T and $A \in K$.
Then

$$\underline{R_A} = \{\langle ft,ft' \rangle \mid \langle t,t' \rangle \in R, f \in Z(T(A)), fX \subseteq A \cup X\}.$$

4.6 Proposition

If R is a well-founded OP-stable relation on T, then R_A is well-founded, too.

Proof:

Suppose that R_A is not well-founded. Then there is a chain $(t_i)_{i \in \mathbb{N}}$ of R_A. By definition of R_A we have two sequences $(u_i)_{i \in \mathbb{N}}$ and

$(u'_i)_{i\in\mathbb{N}}$ of OP-terms as well as a sequence $(f_i)_{i\in\mathbb{N}}$ of S-sorted functions from X to $A\cup X$ such that $f_o u_o = t_o$ and for all $i\in\mathbb{N}$ $<u_i, u'_i>\in R$ and

$$f_i u'_i = f_{i+1} u_{i+1} = t_{i+1}.$$

Therefore, u'_i agrees with u_{i+1} up to some occurrences of elements of $A\cup X$. Hence any S-sorted function $f:X\longrightarrow X$, which satisfies $fx = fy$ whenever $sort(x) = sort(y)$, yields $fu'_i = fu_{i+1}$. Since R is OP-stable, we get $<fu_i, fu'_i>\in R$ for all $i\in\mathbb{N}$. Thus $(fu_i)_{i\in\mathbb{N}}$ is a chain of R contradicting our assumption that R is well-founded. □

We recall the proof principle of Noetherian induction (cf. Cohn /9/, p. 20):

4.7 <u>Proposition</u>

Let M be a set and R a well-founded relation on M. All elements of M satisfy some predicate p if p is derivable by <u>induction w.r.t. R</u>, i.e. if for all $a\in M$ the following condition holds true:
If all $b\in M$ with $<a,b>\in R$ satisfy p, then a satisfies p.

<u>Proof:</u>

Assume that p is derivable by induction w.r.t. R, although some $a_o\in M$ does not satisfy p. Then there must be $a_1\in M$ with $<a_o, a_1>\in R$ and $p(a_1) = false$. Hence $a_2\in M$ exists such that $<a_1, a_2>\in R$ and $p(a_2) = false$. In this way we obtain a chain $(a_i)_{i\in\mathbb{N}}$ of R in contradiction to the well-foundedness of R. □

4.8 Definition

Let $A \in K$. An OP(A)-term t' is a subterm of some $t \in T(A)$ if there are $u \in T$, $x \in var(t)$ and $f \in Z(T(A))$ with $fx = t'$ and $fu = t$. The subterm relation $\supset$ on T(A) consists of all $\langle t,t' \rangle \in T(A)^2$ such that t' is a subterm of t, but different from t. $\subset := \supset^{-1}$. We write $t \supset t'$ or $t' \subset t$ instead of $\langle t,t' \rangle \in \supset$. "$t' \subset t$ or $t' = t$" is abbreviated by $t' \subseteq t$.

4.9 Lemma

Let $A \in K$ and R be a normalizing relation on T(A). Then

$$\underset{R}{>} = (\xrightarrow[R]{} \cup \supset) \cap G(A)$$

is well-founded.

Proof:

Let $t \in T(A)$. We show by induction on t w.r.t. $\xrightarrow[R]{}$ that chains $(t_i)_{i \in \mathbb{N}}$ of $\xrightarrow[R]{} \cup \supset$ with $t_o = t$ do not exist. If $t \in NF(R)$, then all subterms of t are $\xrightarrow[R]{}$-normal, too. Hence t is $(\xrightarrow[R]{} \cup \supset)$-normal. Let $t \notin NF(R)$. Assumed that $(t_i)_{i \in \mathbb{N}}$ is a chain of $\xrightarrow[R]{} \cup \supset$ with $t_o = t$. Then $t_o \supset t_1$ by induction hypothesis. Since $\supset$ is well-founded,

$$t_o \supset t_i \xrightarrow[R]{} t_{i+1}$$

for some $i \geq 1$. Thus there is $t'_i \in T(A)$ with

$$t_o \xrightarrow[R]{} t'_i \supset t_{i+1}$$

in contradiction to the induction hypothesis. Therefore, $\xrightarrow[R]{} \cup \supset$ and thus $\underset{R}{>}$ are well-founded. □

Prop. 4.7 and Lemma 4.9 yield an induction principle we shall refer to in several proofs:

4.10 Corollary

Let $A \in K$ and R be a normalizing relation on $T(A)$. A predicate p holds true for all $t \in G(A)$ if p is provable by induction w.r.t. $>_R$. $\square$

4.11 Completeness Theorem

Let $H \subseteq T^2$ be a term rewriting system such that for all $\langle l,r\rangle \in H$, $A \in K$ and $f \in BZ(G(A))$ $fl \equiv_{SPEC(A)} fr$. If some normalizing and base-complete relation R on T satisfies $H \subseteq R \subseteq H \cup E$, then PAR is complete w.r.t. $\langle BPAR, K\rangle$ (cf. 2.7).

Proof:

Let $A \in K$, $s \in BS$ and $t \in G(A)_s$. The existence of $t' \in BG(A)$ with $t \equiv_{SPEC(A)} t'$ is proved by induction on t w.r.t. $>_R$:
If $t \in NF(R)$, then $t \in BG(A)$ because R is base-complete. Otherwise there is $u \in G(A)$ with $t \xrightarrow[R-H]{} u$ or $t \xrightarrow[H]{} u$. In both cases the induction hypothesis implies $u \equiv_{SPEC(A)} t'$ for some $t' \in BG(A)$.
Therefore, $t \equiv_{SPEC(A)} t'$ in the first case. The second case yields $\langle l,r\rangle \in H$, $f,f',g \in Z(G(A))$, $x \in X$ and $v \in T(A)$ with $var(v) = \{x\}$, $fv = t$, $f'v = u$, $fx = gl$ and $f'x = gr$. Since $l \notin X$, we have $gz \subset gl \subseteq t$ for all $z \in var(l)$. By induction hypothesis, there is $t_z \in BG(A)$ with $gz \equiv_{SPEC(A)} t_z$. Choosing $h \in BZ(G(A))$ with $hz = t_z$ for all $z \in var(l)$ results in $hl \equiv_{SPEC(A)} hr$ by assumption about H. Thus

$$t = fv \equiv_{SPEC(A)} f_o v \equiv_{SPEC(A)} f'_o v \equiv_{SPEC(A)} u$$

where $f_o x = hl$, $f'_o x = hr$ and $f_o z = fz$, $f'_o z = f'z$ for all $z \in X-\{x\}$. Hence we obtain $t \equiv_{SPEC(A)} t'$ in the second case, too. $\square$

4.12 Definition

Let $A \in K$. $t \in T(A)^*$ is linear in $x \in X$ if x occurs in t at most once. t is linear in $Y \subseteq X$ if t is linear in all $x \in Y$. t is linear if t is linear in X. $\langle l,r \rangle \in T(A)^2$ is linear if l is linear. $R \subseteq T(A)^2$ is linear if all $\langle l,r \rangle \in R$ are linear.

4.13 Proposition (Raoult, Vuillemin /58/, Prop. 10)

Let $A \in K$ and R,R' be two linear relations on $T(A)$ such that for all $\langle l,r \rangle \in R$ and $\langle l',r' \rangle \in R'$ l and l' do not overlap in operation symbols.
Let $\xrightarrow[R]{}\!\!\!\!\rightarrow$ denote the least $OP(A)$-stable and parallel $OP(A)$-compatible (cf. 10.5) relation on $T(A)$ that contains R. Then

$$
\begin{tikzcd}
 & u & & u \arrow[dr, ->>, "R'", "\Delta"'] & \\
t \arrow[ur, ->>, "R"] \arrow[dr, ->>, "R'"'] & & \text{implies} & & t' \\
 & u' & & u' \arrow[ur, ->>, "R"', "\Delta"] &
\end{tikzcd}
$$

for some $t' \in T(A)$. $\square$

4.14 Lemma

Let H be a linear, normalizing and base-complete relation on T such that for all $\langle l,r \rangle \in H$

$$op(l) \cap POP = \emptyset.$$

For all $A \in K$ let $HSPEC(A) = \langle S, OP(A), E \cup H \rangle$ and $F_A: G(A) \longrightarrow BG(A)$ such that for all $t,t' \in G(A)$

(i) $t \equiv_{HSPEC(A)} t'$ implies $F_A(t) \equiv_{BSPEC(A)} F_A(t')$,

(ii) $t \in BG(A)$ implies $t \equiv_{BSPEC(A)} F_A(t)$.

Then for all $t,t' \in G(A)$

$$t \equiv_{SPEC(A)} t' \text{ implies } F_A(t) \equiv_{BSPEC(A)} F_A(t').$$

Proof:

Suppose that for all $t,t' \in G(A)$

$$t \equiv_{SPEC(A)} t' \text{ implies } t \xrightarrow[H]{*} \equiv_{BSPEC(A)} \xleftarrow[H]{*} t'. \qquad (*)$$

Then we are done because for all $t \in G(A)$ and $u \in BG(A)$ $t \xrightarrow[H]{*} u$ implies

$$F_A(t) \equiv_{BSPEC(A)} F_A(u) \equiv_{BSPEC(A)} u$$

by (i) and (ii). It remains to show (*).

Let $t \equiv_{SPEC(A)} t'$. There are a least number n and $t_1,\dots,t_n \in G(A)$ with $t_1 = t$, $t_n = t'$ and $t_i \xleftrightarrow[E(A)]{} t_{i+1}$ for all $1 \leq i < n$.
We show

$$t \xrightarrow[H]{*} \equiv_{BSPEC(A)} \xleftarrow[H]{*} t'$$

by induction on n.
Since H is normalizing and base-complete, there is $u \in BG(A)$ with $t \xrightarrow[H]{*} u$.
$n = 1$ implies $t = t'$. Let $n > 1$. By induction hypothesis, $t_2 \xrightarrow[H]{*} u_2 \equiv_{BSPEC(A)} \xleftarrow[H]{*} t'$ for some $u_2 \in BG(A)$.
If $t \xleftrightarrow[E]{} t_2$, then $u \equiv_{HSPEC(A)} u_2$ so that (i) and (ii) imply

$$t \xrightarrow[H]{*} u \equiv_{BSPEC(A)} u_2 \equiv_{BSPEC(A)} \xleftarrow[H]{*} t' .$$

Otherwise $t \xleftrightarrow[\Delta(A)]{} t_2$. Since for all $\langle l,r \rangle \in H$ $op(l) \cap POP = \emptyset$ and $\Delta(A) \in T^2_{POP(A)}$, lefthand sides of H do not overlap lefthand sides of $\Delta(A)$ in operation symbols so that by Prop. 4.13,

$$\begin{tikzcd} & u & & u \arrow[dr, "\Delta(A)", "*"'] & \\ t \arrow[ur, "H", "*"'] \arrow[dr, "\Delta(A)"'] & & \text{implies} & & u' \\ & t_2 & & t_2 \arrow[ur, "*", "H"'] & \end{tikzcd}$$

for some $u' \in G(A)$. Moreover, $u' \in BG(A)$ because $u \in BG(A)$ and $\Delta(A) \subseteq G^2{}_{POP(A)}$. Thus by (i) and (ii),

$$t \xrightarrow[H]{*} u \equiv_{PSPEC(A)} u' \equiv_{BSPEC(A)} u_2 \equiv_{BSPEC(A)} \xleftarrow[H]{*} t'.$$

Hence (*) holds true. □

4.15 <u>Definition</u>

Let $A \in K$ and $R \subseteq T(A)^2$. The pair <u>$\langle R,A \rangle$</u> is <u>absolutely confluent</u> if for all $t,u,u' \in G(A)$

$$\begin{tikzcd} & u & & u \arrow[dr, "R", "*"'] & \\ t \arrow[ur, "R", "*"'] \arrow[dr, "*", "R"'] & & \text{implies} & & t' \\ & u' & & u' \arrow[ur, "*", "R"'] & \end{tikzcd}$$

for some $t' \in G(A)$.

<u>$R \subseteq T^2$</u> is <u>absolutely confluent</u> if for all $A \in K$ $\langle R,A \rangle$ is absolutely confluent.

4.16 <u>Proposition</u>

Let $A \in K$ and R be a normalizing relation on $T(A)$. If $\langle R,A \rangle$ is absolutely confluent, then for all $t \in G(A)$ there is exactly one $t' \in NF(R)$ with $t \xrightarrow[R]{*} t'$. t' is called the <u>normal form of t w.r.t. R</u>, written: <u>$NF_R(t)$.</u>

□

4.17 Definition

Let $A\in K$ and $R\subseteq T(A)^2$. $\langle R,A\rangle$ is base-consistent (w.r.t. BSPEC) if for all $t\in BG(A)$ and $t'\in G(A)$ $t \xrightarrow[R]{} t'$ implies $t \equiv_{BSPEC(A)} t'$.
$R\subseteq T^2$ is base-consistent (w.r.t. BSPEC) if for all $A\in K$ $\langle R,A\rangle$ is base-consistent.

4.18 Lemma

Let $A\in K$ and $R\subseteq T(A)^2$. $\langle R,A\rangle$ is base-consistent iff for all $\langle l,r\rangle\in R$ $l\notin BT(A)$ or for all $f\in Z(G(A))$ $f(var(l))\subseteq BG(A)$ implies $fl \equiv_{BSPEC(A)} fr$.

Proof:

"only if": Let $\langle l,r\rangle\in R$ such that $l\in BT(A)$ and $fl \equiv_{BSPEC(A)} fr$ for some $f\in Z(G(A))$ with $f(var(l))\subseteq BG(A)$. Since $fl\in BG(A)$, but $fl \xrightarrow[R]{} fr$, $\langle R,A\rangle$ is not base-consistent.

"if": Let $t\in BG(A)$ and $t'\in G(A)$ such that $t \xrightarrow[R]{} t'$. Then there are $\langle l,r\rangle\in R$, $f,f',g\in Z(G(A))$, $x\in X$ and $u\in BT(A)$ with $var(u) = \{x\}$, $fu = t$, $f'u = t'$, $fx = gl$ and $f'x = gr$. Since $fu = t\in BG(A)$, $gl = fx$ belongs to $BG(A)$. Therefore, $l\in BT(A)$ and $g(var(l))\subseteq BG(A)$ so that by assumption,

$$fx = gl \equiv_{BSPEC(A)} gr = f'x.$$

Thus $u\in BT(A)$ yields

$$t = fu \equiv_{BSPEC(A)} f'u = t' . \ \square$$

4.19 Consistency Theorem

Let either H be a linear, normalizing and base-complete relation on T such that for all $\langle l,r\rangle\in H$

$op(l)\cap POP = \emptyset$, or let $K = \{\emptyset\}$ and $H = \emptyset$.

For all $A\in K$ let $R(A)$ be a binary relation on $T(A)$ that includes $E\cup H$. If for all $A\in K$ $\langle R(A),A\rangle$ is absolutely confluent and base-consistent, then PAR is consistent w.r.t. <BPAR,K> (cf. 2.7).

<u>Proof:</u>

Let $A \in K$ and $HSPEC(A) = \langle S, OP(A), E \cup H \rangle$. First we show that for all $t, t' \in G(A)$

$$t \equiv_{HSPEC(A)} t' \text{ implies } t \xrightarrow[R(A)]{*} \xleftarrow[R(A)]{*} t' . \qquad (*)$$

Let $t \equiv_{HSPEC(A)} t'$. Then there are a least number n and $t_1, \dots t_n, u_1, \dots u_{n-1} \in G(A)$ with $t_1 = t$, $t_n = t'$ and

$$\begin{tikzcd}
 & u_i \arrow[dl, "*"', "R(A)"] \arrow[dr, "*", "R(A)"'] & \\
t_i & & t_{i+1}
\end{tikzcd}$$

for all $1 \leq i \leq n$. We prove by induction on n that

$$\begin{tikzcd}
t \arrow[dr, "*", "R(A)"'] & & t' \arrow[dl, "*"', "R(A)"] \\
 & v &
\end{tikzcd}$$

for some $v \in G(A)$.

$n = 1$ implies $t = t'$. If $n > 1$, then there are $u, u', v \in G(A)$ with

$$\begin{tikzcd}
t \arrow[dr, "*", "R(A)"'] & & t_2 \arrow[dl, "*"', "R(A)"] \arrow[dr, "*", "R(A)"'] & & t' \arrow[dl, "*"', "R(A)"] \\
 & u \arrow[dr, "*", "R(A)"'] & & u' \arrow[dl, "*"', "R(A)"] & \\
 & & v & &
\end{tikzcd}$$

by absolute confluence of <R(A),A> and induction hypothesis. Hence (*) holds true.

<u>Case 1</u>: H is a linear, normalizing and base-complete relation on T such that for all $<l,r>\in H$ $op(l)\cap POP = \emptyset$. Then there is F_A: $G(A) \longrightarrow BG(A)$ such that for all $t \in G(A)$ $t \xrightarrow[H]{*} F_A(t)$.
Let $t \equiv_{HSPEC(A)} t'$. By (*) and absolute confluence of <R(A),A>, we obtain

$$\begin{tikzcd}
t \arrow[rr, "R(A)"] \arrow[rr, "*"'] \arrow[d, "H"', "*"] & & {} \arrow[dl, "R(A)"', "*"] \arrow[dr, "*"', "R(A)"] & & t' \arrow[ll, "R(A)"'] \arrow[ll, "*"] \arrow[d, "*"', "H"] \\
F_A(t) \arrow[r, "*", "R(A)"'] & {} \arrow[dr, "R(A)"', "*"] & & {} \arrow[dl, "*"', "R(A)"] & F_A(t') \arrow[l, "*"', "R(A)"] \\
& & {} & &
\end{tikzcd}$$

so that 4.14 (i) and (ii) follow from base-consistency of <R(A),A>. Hence by Lemma 4.14,

$$t \equiv_{SPEC(A)} t' \text{ implies } F_A(t) \equiv_{BSPEC(A)} F_A(t'),$$

which results in consistency of PAR by 4.14 (ii).

<u>Case 2</u>: $K = \{\emptyset\}$ and $H = \emptyset$. Then SPEC(A) = HSPEC(A) so that $t \equiv_{SPEC(A)} t'$ with $t,t' \in BG(A)$ implies $t \equiv_{BSPEC(A)} t'$ by (*) and base-consistency of <R(A),A>. Thus PAR is consistent w.r.t. <BPAR,K>.
□

4.20 <u>Corollary</u>

Let either H be a linear, normalizing and base-complete relation on T such that for all $<l,r>\in H$ $op(l)\cap POP = \emptyset$, or let $K = \{\emptyset\}$ and $H = \emptyset$. Let R be a linear and base-consistent relation on T that includes $E \cup H$. If for each two different pairs $\langle l,r \rangle$, $\langle l',r' \rangle \in R$ l and l' do not overlap in operation symbols, then PAR is consistent w.r.t. $\langle BPAR,K \rangle$.

<u>Proof:</u>

By Thm. 4.19 it is sufficient to show that R is absolutely confluent. So let $A \in K$ and

$$\begin{tikzcd}
 & u \\
t \arrow[ur, "*", "R"'] \arrow[dr, "*"', "R"] & \\
 & u'
\end{tikzcd}$$

There are least numbers m and n as also $u_1,\dots,u_m$, $u'_1,\dots,u'_n \in T(A)$ with $u_1 = u'_1 = t$, $u_m = u$, $u'_n = u'$ and for all $1 \leq i \leq m$ resp. $1 \leq i \leq n$

$$u_i \xrightarrow[R]{*} u_{i+1}, \quad u'_i \xrightarrow[R]{*} u'_{i+1}.$$

By induction on n, Prop. 4.13 implies

$$\begin{tikzcd}
u_2 \arrow[dr, "R", "*"'] & \\
 & t_1 \\
u' \arrow[ur, "*", "R"'] &
\end{tikzcd}$$

for some $t_1 \in T(A)$. Induction on m then provides $t_2 \in T(A)$ with

$$\begin{tikzcd}
u \arrow[dr, "*"', "R"] & \\
& t_2 . \\
u' \arrow[ur, "*", "R"'] &
\end{tikzcd}$$

Since $\xrightarrow[R]{*}$ is OP(A)-stable, we get for all $f \in Z(G(A))$

$$\begin{tikzcd}
u = fu \arrow[dr, "*"', "R"] & \\
& ft_2 \\
u' = fu' \arrow[ur, "*", "R"'] &
\end{tikzcd}$$

Thus R is absolutely confluent. $\square$

5. Simplification orderings

This and the following chapter are devoted to sufficient criteria for the normalization property of binary relations R on T. First we shall generalize the Termination Theorem in Dershowitz /11/ in that we prove the well-foundedness of "simplification orderings" (Thm. 5.3). Since simplification orderings have the same closure properties as $\xrightarrow[R]{}$, we thus conclude that R is normalizing whenever it is contained in some simplification ordering. Thm. 5.3 is based on the Tree Theorem in Kruskal /44/ which provides the statement of Lemma 5.1 below only for ground terms. The proof of Lemma 5.1 proceeds along the sketch of a proof of the Tree Theorem given in Dershowitz /11/.

5.1 Lemma

Let $\trianglerighteq$ be the reflexive, transitive and OP-compatible closure of the subterm relation on T (cf.4.8) and $\trianglelefteq = \trianglerighteq^{-1}$. Let $(t_i)_{i\in\mathbb{N}}$ be a sequence of OP-terms such that $\bigcup_{i\in\mathbb{N}} var(t_i)$ is finite. Then there are $i,j\in\mathbb{N}$ with $i<j$ and $t_i\trianglelefteq t_j$.

Proof:

Assume that for all $i,j\in\mathbb{N}$ $i<j$ implies $t_i\ntrianglelefteq t_j$. Since the subterm relation $\supset$ on T is well-founded, we may suppose that $(t_i)_{i\in\mathbb{N}}$ is minimal, i.e.

(*) for all term sequences $(u_i)_{i\in\mathbb{N}}$ such that every $i\in\mathbb{N}$ satisfies $u_i\subset t_j$ for some $j\in\mathbb{N}$ there are $i,j\in\mathbb{N}$ with $i<j$ and $u_i\trianglelefteq u_j$.

First we show that

(**) every such term sequence $(u_i)_{i\in\mathbb{N}}$ even contains a chain of $\trianglelefteq$ as a subsequence.

Let $(u_i)_{i\in\mathbb{N}}$ be a term sequence such that every

$i\in\mathbb{N}$ satisfies $u_i \subset t_j$ for some $j\in\mathbb{N}$. Let I be the set of all $i\in\mathbb{N}$ with $u_i \not\trianglelefteq u_j$ for all $j>i$. By (*), I is finite. Hence there is a subsequence $(u'_i)_{i\in\mathbb{N}}$ of $(u_i)_{i\in\mathbb{N}}$ which is a chain of $\trianglelefteq$. Therefore, (**) holds true.

Since $\bigcup_{i\in\mathbb{N}} var(t_i)$ is finite and since for all $i<j$ $t_i \neq t_j$, there are only finitely many $i\in\mathbb{N}$ with $t_i\in X$. Since OP is also finite, we obtain $n\geq 1, \sigma\in OP$ and a sequence $(u_i)_{i\in\mathbb{N}}$ such that $u_i\in T^n$ and $(\sigma u_i)_{i\in\mathbb{N}}$ is a subsequence of $(t_i)_{i\in\mathbb{N}}$. Let $(u^o_i)_{i\in\mathbb{N}} = (u_i)_{i\in\mathbb{N}}$. By induction on $1\leq k\leq n$ we construct a term tupel sequence $(u^k_i)_{i\in\mathbb{N}}$ which satisfies the following conditions:

(i) $(\sigma u^k_i)_{i\in\mathbb{N}}$ is a subsequence of $(t_i)_{i\in\mathbb{N}}$,

(ii) for all $1\leq m\leq k$ $(u^k_{i,m})_{i\in\mathbb{N}}$ is a chain of $\trianglelefteq$.

Let $0\leq k<n$, and let $(u^k_i)_{i\in\mathbb{N}}$ be a term tupel sequence with (i) and (ii). By (i) and (**), for all $1\leq j\leq n$ there is a subsequence $(u'_{i,j})_{i\in\mathbb{N}}$ of $(u^k_{i,j})_{i\in\mathbb{N}}$ which is a chain of $\trianglelefteq$. We set

$$(u_i^{k+1})_{i\in\mathbb{N}} = (u'_i)_{i\in\mathbb{N}}.$$

By (i), $(\sigma u_i^{k+1})_{i\in\mathbb{N}}$ is a subsequence of $(t_i)_{i\in\mathbb{N}}$, and by (ii) and transitivity of $\trianglelefteq$, for all $1\leq m\leq k$ $(u_i^{k+1})_{i\in\mathbb{N}}$ is a chain of $\trianglelefteq$. Furthermore, for all $i\in\mathbb{N}$

$$u^{k+1}_{i,k+1} = u'_{i,k+1} \trianglelefteq u'_{i+1,k+1} = u^{k+1}_{i+1,k+1}$$

so that $(u^{k+1}_{i,k+1})_{i\in\mathbb{N}}$ is also a chain of $\trianglelefteq$.

Hence (i) and (ii) hold true for k+1 instead of k. Since $\trianglelefteq$ is OP-compatible, (ii) implies $\sigma u_o{}^n \trianglelefteq \sigma u_1{}^n$ which, by (i), contradicts our assumption that all $i<j$ satisfy $t_i \ntrianglelefteq t_j$. □

5.2 Definition

A binary relation R on a set M is irreflexive if for all $a \in M$ $\langle a,a \rangle \notin R$.
An irreflexive, transitive, OP-compatible and OP-stable relation on T that contains the subterm relation on T is called a simplification ordering.

5.3 Theorem

Simplification orderings are well-founded.

Proof:

Let R be a simplification ordering. If R would not be well-founded, there would be a chain $(t_i)_{i \in N}$ of R. Let $f: X \longrightarrow X$ be an S-sorted function such that for all $x,y \in X$ sort(x) = sort(y) implies fx = fy. Since S is finite, $\bigcup_{i \in N} var(ft_i)$ is finite, too. By Thm. 5.1, there are $i<j$ with $ft_i \trianglelefteq ft_j$. Since R is transitive and OP-stable, $\langle ft_i, ft_j \rangle \in R$. By irreflexivity of R, $ft_i \triangleleft ft_j$ where $\trianglelefteq$ denotes the transitive and OP-compatible closure of $\subset$.
Next we show that $\triangleleft$ is irreflexive.
Since the relation $\prec \subseteq T^2$ defined by

$$t \prec t' \text{ iff } /op(t)/ < /op(t')/$$

is transitive and OP-compatible and contains $\subset$, it also includes $\triangleleft$. Therefore, the irreflexivity of $\prec$ implies that $\triangleleft$ is irreflexive, too.
Hence R contains $\triangleright = \triangleleft^{-1}$ so that $\langle ft_j, ft_i \rangle \in R$ and thus $\langle ft_j, ft_j \rangle \in R$ by transitivity of R,

contradicting the assumption that R is irreflexive. Hence R is well-founded. □

5.4 Corollary

Let R be a simplification ordering.

(i) All subsets of R are normalizing.

(ii) Let $A \in K$. All subsets of R_A are normalizing (cf. 4.5).

Proof:

(i) Let $R' \subseteq R$. Then $\xrightarrow[R']{} \subseteq R$ so that by Thm. 5.3, R' is normalizing.

(ii) Prop. 4.6 carries over the well-foundedness from R to R_A. Let $R' \subseteq R_A$. Provided that R_A is OP(A)-compatible and -stable we have $\xrightarrow[R']{} \subseteq R_A$ implying that R' is normalizing. We first show that R_A is OP(A)-compatible. Let $w \in S^+$, $\sigma \in OP(A)$, $t,t' \in T(A)_w$, $\langle t_j, t'_j \rangle \in R_A$ for some $1 \leq j \leq n = lg(w)$ and $t_i = t'_i$ for all $1 \leq i \leq n$ with $i \neq j$. For all $1 \leq i \leq n$ there are $\langle u_i, u'_i \rangle \in R$ and $f_i \in Z(T(A))$ with $f_i X \subseteq A \cup X$, $f_i u_i = t_i$ and $f_i u'_i = t'_i$. Furthermore, let $g_i : X \longrightarrow X$, $1 \leq i \leq n$, be injective S-sorted functions such that the sets $X_i = var(g_i u_i) \cup var(g_i u'_i)$ are pairwise disjoint. We define $f \in Z(T(A))$ by

$$fx = \begin{cases} f_i g_i^{-1} x & \text{if } x \in X_i \\ x & \text{otherwise} \end{cases}$$

and obtain

$$\sigma(t_1,\ldots,t_n) = \sigma(f_1u_1,\ldots,f_nu_n)$$
$$= \sigma(fg_1u_1,\ldots,fg_nu_n)$$
$$= f\sigma(g_1u_1,\ldots,g_nu_n) \qquad (*)$$

and analogously,

$$\sigma(t'_1,\ldots,t'_n)$$
$$= f\sigma(g_1u'_1,\ldots,g_nu'_n). \qquad (**)$$

Since R is OP-stable and OP-compatible, $\langle u_i,u'_i\rangle \in R$ implies

$$\langle\sigma(g_1u_1,\ldots,g_nu_n),\sigma(g_1u'_1,\ldots,g_nu'_n)\rangle$$

Thus by (*) and (**),

$$\langle\sigma(t_1,\ldots,t_n),\sigma(t'_1,\ldots,t'_n)\rangle \in R_A.$$

Therefore, R_A is OP(A)-compatible.

OP(A)-stability of R_A holds true as follows:
Let $\langle t,t'\rangle \in R_A$ and $g \in Z(T(A))$. Then there are $\langle u,u'\rangle \in R$ and $f \in Z(T(A))$ with $fX \subseteq A \cup X$, $fu = t$ and $fu' = t'$.
Let $M = \{x \in var(u) \cup var(u') / fx \in X\}$.
There are $f'\colon X \longrightarrow A \cup X$ and $t_x \in M$ for all $x \in M$ such that $f't_x = gfx$ and the sets $var(t_x)$, $x \in M$, are pairwise disjoint as well as disjoint from $var(u) \cup var(u')$. We define $g' \in Z(T)$ and $f''\colon X \longrightarrow A \cup X$ by

$$g'x = \begin{cases} t_x & \text{if } x \in M \\ x & \text{otherwise} \end{cases}$$

resp.

$$f''x = \begin{cases} f'x \text{ if } x \in var(t_y),\ y \in M \\ fx \text{ otherwise} \end{cases}$$

and obtain for all $x \in M$

$$gfx = f't_x = f''t_x = f''g'x$$

and for all $x \in (var(u) \cup var(u')) - M$

$$gfx = fx = f''x = f''g'x.$$

Hence $gt = gfu = f''g'u$ and

$gt' = gfu' = f''g'u'$.

Since R is OP-stable, $\langle u,u'\rangle \in R$ implies

$\langle g'u,g'u'\rangle \in R$.

Thus $\langle gt,gt'\rangle = \langle f''g'u,f''g'u'\rangle \in R$.

Therefore, R_A is OP(A)-stable. □

Corollary 5.4 (i) and Thm. 4.11 yield

5.5 Completeness Theorem

Let $H \subseteq T^2$ be a term rewriting system such that for all $\langle l,r\rangle \in H$, $A \in K$ and $f \in BZ(G(A))$ $fl \equiv_{SPEC(A)} fr$.
If some base-complete relation R on T and some simplification ordering R' satisfy $R \subseteq R'$ and $H \subseteq R \subseteq H \cup E$, then PAR is complete w.r.t. $\langle BPAR,K\rangle$. □

6. Recursively decreasing term relations

This chapter deals with an important class of simplification orderings, the so-called recursive path orderings introduced in Plaisted /57/ and Dershowitz /12/. Recursive path orderings are inductively defined extensions of the subterm relation on T which depend - in their form described below - on two binary relations O_1 and O_2 on OP resp. S. Term relations which are contained in recursive path orderings will be called recursively decreasing. If such relations occur in the context of equational specifications, they are mostly directly decreasing in the sense of Def. 6.8. A decision procedure for "directly decreasing" is much faster than one for "recursively decreasing" because the definition of "directly decreasing" is non-recursive.

The recursion in the definition of a recursive path ordering requires an extension of well-founded term relations to well-founded term tupel relations. For that purpose we have chosen a lexicographic extension which does not depend on the length of the term tupels to be compared (cf. 6.1). If we would demand a term tupel t to be not "greater" than a term tupel t' in the case that t is shorter than t', such an order would be useless for a completeness proof of a specification with "mutual-recursive" operations of different rank. Moreover, the lexicographic extension defined here admits to prove the normalization property of equations with nested recursion. Kamin, Levy /39/ have shown that this cannot be done with the help of a recursive path ordering which uses a multiset extension (cf. Dershowitz, Manna /13/). Every recursive path ordering with lexicographic extension can be transformed into a recursive path ordering with multiset extension (Pettorossi /56/, Thm. 8), but the multisets generated by that transformation do not simply consist of term tupel components as in Dershowitz /12/, rather they are derived from the term tupels in a more or less complicated way. In order to get around such additional constructions we restrict ourselves to recursive path orderings with lexicographic extension, especially as we do not know specifications which demand greater generality.

6.1 Definition

Let $k = \max\{\mathrm{rank}(\sigma)/\sigma \in OP\}$ and R, O be two binary relations on T resp. S. The relation

$$\underline{LEX(R,O)} \subseteq \bigcup_{i=1}^{k} T^i$$

defined as follows is called the lexicographic extension of R w.r.t. O:

Let $w,w' \in S^*$, $t \in T_w$ and $t' \in T_{w'}$. $\langle t,t'\rangle \in LEX(R,O)$ if there are $1 \leq i \leq \min(lg(w),lg(w'))$ with $t_j = t'_j$ for all $1 \leq j < i$ and

(i) $\langle t_i, t'_i\rangle \in R$ and $\langle w_i, w'_i\rangle \in O^*$

or

(ii) $\langle w_i, w'_i\rangle \in O^+$ and $\langle w'_i, w_i\rangle \notin O^*$.

6.2 Lemma

If $R \subseteq T^2$ is well-founded, so are all lexicographic extensions of R.

Proof:

Let $O \subseteq S^2$. O induces the following equivalence relation $\sim$ on S: $s \sim s'$ iff $\langle s,s'\rangle \in O^* \cap (O^*)^{-1}$. Since R is well-founded, so are all restrictions $R_{[s]}$, $s \in S$, on $T_{[s]} = \bigcup_{s' \sim s} T_{s'}$ where [s] denotes the equivalence class of s w.r.t. $\sim$. The disjointness of all $T_{[s]}$ implies that

$$R_o = \Big(\bigcup_{s\in S} R_{[s]}\Big) \cup \Big(\bigcup_{\substack{\langle s,s'\rangle \in O^+ \\ s \not\sim s'}} T_{[s]} \times T_{[s]}\Big)$$

is well-founded, too.

Let $R' = LEX(R,O)$. For all $\langle t,t'\rangle \in R'$ let $f(t,t')$ be the index of 6.1 (i) resp. (ii). Suppose that $(t_i)_{i\in \mathbb{N}}$ is a chain of R' . Since for all i $lg(t_i) \leq k$, the set

$$M((t_i)_{i\in\mathbb{N}}) = \{f(t_i, t_{i+1}) / i\in\mathbb{N}\}$$

is finite. We show by induction on $n = /M((t_i)_{i\in\mathbb{N}})/$ that R_o has a chain in contradiction to the proposition above.

If $n = 1$, then there is $1 \leq j \leq k$ such that for all $i\in\mathbb{N}$ $j = f(t_i, t_{i+1})$. Hence $\langle t_{i,j}, t_{i+1,j}\rangle \in R_o$ for all $i\in N$, i.e. $(t_{i,j})_{i\in\mathbb{N}}$ is a chain of R_o.

If $n>1$, then let $j = \min(M((t_i)_{i\in\mathbb{N}}))$ and

$$I = \{i\in\mathbb{N} / f(t_i, t_{i+1}) = j\}.$$

For all $i\in I$ $\langle t_{i,j}, t_{i+1,j}\rangle \in R_o$, while for all $i\in\mathbb{N}-I$ $t_{i,j} = t_{i+1,j}$. Therefore, if I is infinite, then $(t_{i,j})_{i\in I}$ is a chain of R_o.

Otherwise

$$M((t_i)_{i>\max(I)}) = M((t_i)_{i\in\mathbb{N}}) - \{j\}$$

so that $/M((t_i)_{i>\max(I)})/ < n$, and by induction hypothesis, R_o has a chain. $\square$

6.3 <u>Definition</u>

Let O_1 and O_2 be two binary relations on OP resp. S The relation $R = \underline{REC(O_1, O_2)} \subseteq T^2$ defined inductively as follows is called <u>recursive path ordering w.r.t. O_1 and O_2</u>:

(i) For all $t,u\in T$, $t \supset u$ and $\langle u,v\rangle \in R^{\Delta}$ implies $\langle t,v\rangle\in R$.

(ii) For all $w,w'\in S^*$, $t\in T_w, u\in T_{w'}$ and $\sigma,\tau\in OP$ with $arity(\sigma) = w$ and $arity(\tau) = w'$

(a) $\langle\sigma,\tau\rangle\in O_1{}^*$,

(b) $\langle\sigma t,u_i\rangle\in R$ for all $1\leq i\leq lg(w')$ and

(c) $\langle\sigma,\tau\rangle\in O_1{}^*$ or $\langle t,u\rangle\in LEX(R,O_2)$

imply $\langle\sigma t,\tau u\rangle\in R$.

6.4 Example

Let $S = \{s\}$,

$OP = \{+:ss\longrightarrow s, \sigma:s\longrightarrow s\}$,

$O_1 = \{\sigma,+\}$

$O_2 = \emptyset$.

The rule (cf. 4.4)

$$\langle\sigma(x+\sigma y),\sigma(x+y)+\sigma y\rangle$$

belongs to $R = REC(O_1,O_2)$: Since x and y are subterms of $\sigma(x+\sigma y)$, 6.3 (i) implies

$$\langle\sigma(x+\sigma y),x\rangle, \langle\sigma(x+\sigma y),y\rangle\in R. \qquad (1)$$

By 6.3 (i), $\langle\sigma y,y\rangle\in R$ so that

$$\langle\langle x,\sigma y\rangle,\langle x,y\rangle\rangle\in LEX(R,O_2). \qquad (2)$$

Moreover,

$$\langle x+\sigma y,x\rangle, \langle x+\sigma y,y\rangle\in R. \qquad (3)$$

$\langle\sigma,+\rangle\in O_1$, (2) and (3) imply 6.3 (ii), (a) - (c) so that

$$\langle x+\sigma y,x+y\rangle\in R\subseteq LEX(R,O_2). \qquad (4)$$

$\langle\sigma,\sigma\rangle\in O_1{}^*$, (1) and (4) yield 6.3 (ii), (a) - (c) and thus

$$\langle\sigma(x+\sigma y),\sigma(x+y)\rangle\in R, \qquad (5)$$

while

$$\langle\sigma(x+\sigma y), \sigma y\rangle\in R \qquad (6)$$

follows from 6.3 (i). Finally, (5), (6) and $\langle +,\sigma\rangle\notin O_1{}^*$ imply

$$\langle\sigma(x+\sigma y),\sigma(x+y)+\sigma y\rangle\in R$$

by 6.3 (ii).

6.5 Theorem

Let $O_1 \subseteq OP^2$ and $O_2 \subseteq S^2$. $REC(O_1,O_2)$ is a simplification ordering.

Proof:

We proceed along the corresponding proof for recursive path orderings on ground terms with multiset extension given in Dershowitz /12/ (Thm. 3; see the introduction to this chapter).

Let $R = REC(O_1,O_2)$. We first show that R is transitive: Let $\langle t,u\rangle,\langle u,v\rangle \in R$. $\langle t,v\rangle \in R$ follows by induction on size(t) + size(u) + size(v).

Case 1: size(t) = size(u) = 1. By definition of R, t and u are constants with $\langle t,u\rangle,\langle u,\mathrm{root}(v)\rangle \in O_1{}^*$ and $\langle \mathrm{root}(v),u\rangle,\langle u,t\rangle \notin O_1{}^*$. Hence $\langle t,\mathrm{root}(v)\rangle$ and $\langle \mathrm{root}(v),t\rangle \in O_1{}^*$. If size(v) = 1, then $\langle t,v\rangle \in R$ by 6.3 (ii). Otherwise by definition of R, there are $\sigma \in OP$, $n \geq 1$ and $v' \in T^n$ such that $\sigma v' = v$ and for all $1 \leq i \leq n$ $\langle u,v'_i\rangle \in R$. By induction hypothesis, $\langle t,v'_i\rangle \in R$ for all $1 \leq i \leq n$. By 6.3 (ii), $\langle t,\sigma\rangle \in O_1{}^*$ and $\langle \sigma,t\rangle \notin O_1{}^*$ results in $\langle t,v\rangle \in R$.

Case 2: size(t) + size(u) > 2. By definition of R, we have to distinguish between three subcases:

Case 2.1: $t \supset u$. Then $\langle t,v\rangle \in R$ by 6.3(i).

Case 2.2: There is t' with $t \supset t'$ and $\langle t',u\rangle \in R$. By induction hypothesis, we obtain $\langle t',v\rangle \in R$ so that $\langle t,v\rangle \in R$ by 6.3(i).

Case 2.3: There are $\sigma,\tau \in OP$, $n,m \in N$, $t' \in T^n$ and $u' \in T^m$ such that $\sigma t' = t$, $\tau u' = u$, $\langle \sigma,\tau\rangle \in O_1{}^*$ and for all $1 \leq i \leq m$, $\langle t,u_i'\rangle \in R$. Again we have three subcases:

Case 2.3.1: $u \supset v$. Then $u_i' \supseteq v$ for some $1 \leq i \leq m$. If $u_i' = v$, then $\langle t,v \rangle = \langle t,u_i' \rangle \in R$. Otherwise $\langle u_i', v \rangle \in R$ by 6.3(i), and we obtain $\langle t,v \rangle \in R$ by induction hypothesis.

Case 2.3.2: There are u'' with $u \supset u''$ and $\langle u'', v \rangle \in R$. Then $u_i' \supseteq u''$ for some $1 \leq i \leq m$. If $u_i' = u''$, $\langle t,v \rangle \in R$ follows by induction hypothesis. Otherwise $\langle u_i', v \rangle \in R$ by 6.3(i), and the induction hypothesis implies $\langle t,v \rangle \in R$.

Case 2.3.3: There are $\varphi \in OP$, $r \in \mathbb{N}$ and $v' \in T^r$ such that $\varphi v' = v$, $\langle \tau, \varphi \rangle \in O_1{}^*$ and for all $1 \leq i \leq r$, $\langle u, v_i' \rangle \in R$. By induction hypothesis, $\langle t, v_i' \rangle \in R$ for all $1 \leq i \leq r$. If $\langle \varphi, \sigma \rangle \in O_1{}^*$, then $\langle t,v \rangle \in R$ follows from 6.3(ii). Otherwise $\langle \varphi, \tau \rangle, \langle \tau, \sigma \rangle \in O_1{}^*$, which implies

$$\langle t',u' \rangle, \langle u',v' \rangle \in LEX(R,O_2) \qquad (*)$$

since we may assume that cases 2.1, 2.2, 2.3.1 and 2.3.2 do not hold true. By Definition 6.1, there are $1 \leq i \leq \min(\mathrm{rank}(\sigma),\mathrm{rank}(\tau))$ and $1 \leq j \leq \min(\mathrm{rank}(\tau),\mathrm{rank}(\varphi))$ such that for all $1 \leq s < i$ $t_s' = u_s'$, for all $1 \leq s < j$ $u_s' = v_s'$ and such that one of the following cases occurs:

Case 2.3.3.1: $i<j$. Then $t_s' = u_s' = v_s'$ for all $1 \leq s < i$, and $u_i' = v_i'$. If 6.1(i) holds true for $\langle t_i', u_i' \rangle$, i.e. $\langle t_i', u_i' \rangle \in R$ and $\langle \mathrm{sort}(t_i'), \mathrm{sort}(u_i') \rangle \in O_2{}^*$, we obtain $\langle t_i', v_i' \rangle \in R$ and $\langle \mathrm{sort}(t_i'), \mathrm{sort}(v_i') \rangle$ O_2^*.
If 6.1(ii) is valid, i.e.
$\langle \mathrm{sort}(t_i'), \mathrm{sort}(u_i') \rangle \in O_2{}^+ - (O_2{}^*)^{-1}$, then
$\langle \mathrm{sort}(t_i'), \mathrm{sort}(v_i') \rangle \in O_2{}^+ - (O_2{}^*)^{-1}$.

Case 2.3.3.2: $i \geq j$. Then $t_s' = u_s' = v_s'$ for all $1 \leq s < j$, and $\langle t_j', u_j' \rangle \in R^{\Delta}$. Hence by (*) and 6.1(i), (ii), $\langle \mathrm{sort}(t_j'), \mathrm{sort}(u_j') \rangle \in O_2{}^*$. If 6.1(i) holds true for $\langle u_j', v_j' \rangle$, i.e. $\langle u_j', v_j' \rangle \in R$ and $\langle \mathrm{sort}(u_j'), \mathrm{sort}(v_j') \rangle \in O^*{}_2$,

we obtain $\langle t_j', v_j'\rangle \in R$ by induction hypothesis, and $\langle sort(t_j'), sort(v_j')\rangle \in O_2^*$. If 6.1(ii) is valid, i.e.
$\langle sort(u_j'), sort(v_j')\rangle \in O_2^+ - (O_2^*)^{-1}$,
then
$\langle sort(t_j'), sort(v_j')\rangle \in O_2^+ - (O_2^*)^{-1}$.
($\langle sort(v_j'), sort(t_j')\rangle \in O_2^*$ would contradict
$\langle sort(v_j'), sort(u_j')\rangle \in O_2^*$.)

Therefore, in both cases 6.1 implies $\langle t', v'\rangle \in LEX(R, O_2)$ so that $\langle t, v\rangle \in R$ by 6.3(ii).

<u>R is irreflexive</u>: We show $\langle t, t\rangle \notin R$ by induction on size(t). If size(t) = 1, $\langle t, t\rangle \notin R$ follows from the definition of R. Assume that size(t)>1 and $\langle t, t\rangle \in R$. Then we have one of the following two cases:

<u>Case 1</u>: There is u with $t \supset u$ and $\langle u, t\rangle \in R$. By 6.3(i), $\langle t, u\rangle \in R$ so that transitivity of R implies $\langle u, u\rangle \in R$ - in contradiction to the induction hypothesis.

<u>Case 2</u>: There are $\sigma \in OP$, $n \geq 1$ and $u \in T^n$ with $\sigma u = t$ and $\langle u, u\rangle \in LEX(R, O_2)$. This contradicts the induction hypothesis because there must be $1 \leq i \leq n$ with $\langle u_i, u_i\rangle \in R$.

<u>R is OP-compatible</u>: Let $w \in S^+$, $s \in S$, $\sigma \in OP_{w,s}$, $t, u \in T_w$ and $1 \leq j \leq n = lg(w)$ such that $\langle t_j, t_j'\rangle \in R$ and for all $1 \leq i \leq n$ with $i \neq j$, $t_i = t_i'$. 6.3(i) implies $\langle \sigma t, t_i'\rangle \in R$ for all $1 \leq i \leq n$. Furthermore, $\langle t, t'\rangle \in LEX(R, O_2)$ so that $\langle \sigma t, \sigma t'\rangle \in R$ by 6.3(ii).

<u>R is OP-stable</u>: Let $\langle t, u\rangle \in R$ and $f \in Z(T)$. We show $\langle ft, fu\rangle \in R$ by induction on size(t) + size(u).

<u>Case 1</u>: size(t) = size(u) = 1. By definiton of R, t and u are constants. Hence ft = t and fu = u so that $\langle ft, fu\rangle \in R$.

Case 2: size(t) = 1 and size(u)>1. By definition of R, $t \in OP$ and there are $\sigma \in OP, n \geq 1$ and $u' \in T^n$ such that $\sigma u' = u$, $<t,\sigma> \in O_1{}^*$, $<t,u_i'> \in R$ for all $1 \leq i \leq n$, and $<\sigma,t> \notin O_1{}^*$. By induction hypothesis, $<ft,fu_i'> \in R$ for all $1 \leq i \leq n$. Since ft = t and root(fu) = σ, we obtain $<ft,fu> \in R$ by 6.3(ii).

Case 3: size(t)>1.

Case 3.1: There are t' with $t \supset t'$ and $<t',u> \in R^{\Delta}$. Then $ft \supset ft'$ and by induction hypothesis, $<ft',fu> \in R^{\Delta}$, which implies $<ft,fu> \in R$ by 6.3(i).

Case 3.2: There are $n \geq 1$, $m \geq 0$, $t' \in T^n$, $u' \in T^m$ and $\sigma,\tau \in OP$ with $\sigma t' = t, \tau u' = u$, $<\sigma,\tau> \in O_1{}^*$, $<t,u_i'> \in R$ for all $1 \leq i \leq m$, and

$<\tau,\sigma> \notin O_1{}^*$ or $<t',u'> \in LEX(R,O_2)$.

By induction hypothesis, $<ft,fu_i'> \in R$ for all $1 \leq i \leq m$.

Case 3.2.1: size(u) = 1, i.e. $u' = \varepsilon$. Then by definition of $LEX(R,O_2)$, $<t',u'> \in LEX(R,O_2)$ and thus $<\tau,\sigma> \notin O_1{}^*$. Hence $<ft,fu> = <\sigma ft',\tau> \in R$ by 6.3(ii).

Case 3.2.2: size(u)>1. If $<\tau,\sigma> \in O_1{}^*$, then $<ft,fu> = <\sigma ft',\tau fu'> \in R$ by 6.3(ii). If $<t',u'> \in LEX(R,O_2)$, then there are $w,w' \in S^*$ and $1 \leq i \leq \min(lg(w),lg(w'))$ with $t' \in T_w$, $u' \in T_{w'}$, $t_j' = u_j'$ for all $1 \leq j < i$ such that one of the following two cases holds true.

Case 3.2.2.1: $<t_i',u_i'> \in R$ and $<w_i,w_i'> \in O_2{}^*$. By induction hypothesis, $<ft_i',fu_i'> \in R$ so that $ft' \in T_w$ and $fu' \in T_{w'}$ imply $<ft',fu'> \in LEX(R,O_2)$.

Case 3.2.2.2: $<w_i,w_i'> \in O_2{}^+ - (O_2{}^*)^{-1}$. Again, $ft' \in T_w$ and $fu' \in T_{w'}$ imply $<ft',fu'> \in LEX(R,O_2)$.

Hence both cases result in $\langle ft,fu\rangle = \langle ft,\tau fu'\rangle \in R$.

Finally, 6.3(i) implies that R contains the subterm relation on T. □

6.6 Definition

A binary relation R on T is recursively decreasing if $R \subseteq REC(O_1,O_2)$ for some $O_1 \subseteq OP^2$ and $O_2 \subseteq S^2$.

Thms. 6.5 and 5.5 yield

6.7 Completeness Theorem

Let $H \subseteq T^2$ be a term rewriting system such that for all $\langle l,r\rangle \in H$, $A \in K$ and $f \in BZ(G(A))$ $fl \equiv_{SPEC(A)} fr$. If some recursively decreasing and base-complete relation R satisfies $H \subseteq R \subseteq H \cup E$, then PAR is complete w.r.t. $\langle BPAR,K\rangle$. □

6.8 Definition

Let R be a term rewriting system on T, $O_2(R) \subseteq S^2$, and let $\underline{O_1(R)} \subseteq OP^2$ be defined as follows:

$\langle \sigma,\tau\rangle \in O_1(R)$ iff there is $\langle l,r\rangle \in R$ with $root(l) = \sigma$ and $\tau \in op(r)$.

R is directly decreasing if for all $\langle l,r\rangle \in R$ and all subterms t of r, $\langle root(t),root(l)\rangle \in O_1(R)$ implies

$\langle arg(l),arg(t)\rangle \in LEX(\supset,O_2(R))$. (*)

6.9 Example (Ackermann function)

Let

SPEC = nat +

opns: ACK: nat nat $\longrightarrow$ nat

eqns: ACK(0,x) = Sx e1

ACK(Sx,O) = ACK(x,SO) e2

ACK(Sx,Sy) = ACK(x,ACK(Sx,y)) e3

Choosing $R = \{$e1, e2, e3$\}$, we obtain

$$O_1(R) = \{<ACK,\tau>/\ \tau\in\{0,S,ACK\}\}$$

and set $O_2(R) = \emptyset$.

R is directly decreasing: The righthand side of e1 does not contain subterms t with $<root(t),ACK>\in O_1(R)^*$. The only subterm in the righthand side r of e2 that satisfies this property is r. Since $<<Sx,O>,<x,SO>>\in LEX(\supset,\emptyset)$, 6.8(*) holds true for $<l,r>$ = e2 and t = r. As to $<l,r>$ = e3, there are two subterms t of r with $<root(t),ACK>\in O_1(R)^*$, namely t = ACK(x,ACK(Sx,y)) and t = ACK(Sx,y). Since $<<Sx,Sy>,<x,ACK(Sx,y)>>$ and $<<Sx,Sy>,<Sx,y>>$ belong to $LEX(\supset,\emptyset)$, both cases imply 6.8(*). Not all recursively decreasing relations are directly decreasing: If R consists of the rule $<l,r>$ given in Example 6.4, we have $O_1(R) = \{<\sigma,+>,<\sigma,\sigma>\}$. Since $\sigma(x+y)$ is a subterm t of r satisfying $<root(t),root(l)>\in O_1(R)^*$ as well as

$$arg(l) \not\supset arg(t),$$

R is not directly decreasing. But vice versa, we have

6.10 Theorem

Directly decreasing term rewriting systems are recursively decreasing.

Proof:

Let R be a directly decreasing relation. Then $\hat{R} = \{<l,t> \in T^2 / t \leq r$ for some $<l,r> \in R$ is directly decreasing, too - with $O_1(\hat{R}) = O_1(R)$. Let $<l,r> \in \hat{R}$. We show by induction on the maximal nesting degree $n(r)$ of subterms t of r with $<root(t),root(l)> \in O_1(R)^*$ that $<l,r>$ belongs to $\bar{R} = REC(O_1(R),O_2(R))$. Let $n(r) = 0$. Since $<l,r>$ is a rule, for all $x \in var(r)$ $l \supset x$ and thus $<l,x> \in \bar{R}$. $n(r) = 0$ implies $<\tau,root(l)> \notin O_1(R)^*$ for all $\tau \in op(r)$. Since $<l,r> \in \hat{R}$, $<root(l),\tau> \in O_1(R)$ for all $\tau \in op(r)$. Hence repeated applications of 6.3(ii) imply $<l,r> \in \bar{R}$.

Let $n(r) > 0$ and M be the set of all subterms t of r with $<root(t),root(l)> \in O_1(R)^*$. Let $t \in M$. Then there are $\tau \in OP$, $m \in \mathbb{N}$ and $u \in T^m$ such that $t = \tau u$. For all $1 \leq i \leq m$, $<l,u_i> \in \hat{R}$ and $n(u_i) < n(t) \leq n(r)$. By induction hypothesis, $<l,u_i> \in \bar{R}$ for all $1 \leq i \leq m$. Since $\bar{R}$ includes $\supset$,

$$<arg(l),u> \in LEX(\supset,O_2(\hat{R})) \subseteq LEX(\bar{R},O_2(\hat{R})) = LEX(\bar{R},O_2(R)).$$

Therefore, $<l,t> \in \bar{R}$ by 6.3(ii). Since for all $\tau \in op(r) - \{root(t)/t \in M\}$ $<\tau,root(l)> \notin O_1(R)^*$, repeated applications of 6.3(ii) yield $<l,r> \in \bar{R}$. □

Thms. 6.10 and 6.7 result in

6.11 Completeness Theorem

Let $H \subseteq T^2$ be a term rewriting system such that for all $<l,r> \in H, A \in K$ and $f \in BZ(G(A))$ $fl \equiv_{SPEC(A)} fr$. If some directly decreasing and base-complete relation R satisfies $H \subseteq R \subseteq H \cup E$, then PAR is complete w.r.t. $<BPAR,K>$. □

The following example should illustrate that even complex term rewriting systems arising in practice are directly decreasing. The example is taken from the specification of an interpreter for imperative programs with assignment (ASSIGN), composition (COMP) and bounded iteration (LOOP). The specification was first presented in Klären, Petzsch /41/ as a "structural-recursive schema".

6.12 Example

Let

SPEC = nat +

sorts: ident, expr, stmt, stmtl, store

opns: IDE: ident ⟶ expr

ASSIGN: ident expr ⟶ stmt

SIMPL : stmt ⟶ stmtl

COMP: stmt stmtl ⟶ stmtl

LOOP: ident stmtl ⟶ stmt

PUT: store ident nat ⟶ store

EXPR: expr store ⟶ nat

STMTL: stmtl store ⟶ store

STMT: stmt store ⟶ store

ITERATE: stmtl nat store ⟶ store

eqns: STMTL(SIMPL(stm),sto) = STMT(stm,sto) e1

STMTL(COMP(stm,stml),sto)
= STMTL(stml,STMT(stm,sto)) e2

STMT(ASSIGN(x,e),sto)
= PUT(sto,x,EXPR(e,sto)) e3

STMT(LOOP(x,stml),sto)
= ITERATE(stml,EXPR(IDE(x),sto),sto) e4

ITERATE(stml,0,sto) = sto e5

ITERATE(stml,Sn,sto)
= ITERATE(stml,n,STMTL(stml,sto)) e6

Choosing R = {e1,...,e6}, we obtain

$O_1(R) = \{\langle STMTL,STMT\rangle, \langle STMTL,STMTL\rangle, \langle STMT,PUT\rangle, \langle STMT,EXPR\rangle, \langle STMT,ITERATE\rangle, \langle STMT,IDE\rangle, \langle ITERATE,ITERATE\rangle, \langle ITERATE,STMTL\rangle\}$.

The following table lists for each equation $\langle l,r\rangle \in R$ all subterms t of r with $\langle root(t),root(l)\rangle \in O_1(R)^*$:

<u>e1</u> STMT(stm,sto)
<u>e2</u> STMTL(stml,STMT(stm,sto)),STMT(stm,sto)
<u>e3</u> -
<u>e4</u> ITERATE(stml,EXPR(IDE(x),sto),sto)
<u>e5</u> -
<u>e6</u> ITERATE(stml,n,STMTL(stml,sto)),STMTL(stml,sto)

Hence R is directly decreasing if the following term tupel pairs belong to $LEX(\supset,O_2(R))$ where $O_2(R) = \{\langle \underline{nat},\underline{store}\rangle\}$:

(a) $\langle\langle SIMPL(stm),sto\rangle,\langle stm,sto\rangle\rangle$
(b) $\langle\langle COMP(stm,stl),sto\rangle,\langle stml,STMT(stm,sto)\rangle\rangle$
(c) $\langle\langle COMP(stm,stl),sto\rangle,\langle stm,sto\rangle\rangle$
(d) $\langle\langle LOOP(x,stml),sto\rangle,\langle stml,EXPR(IDE(x),sto),sto\rangle\rangle$
(e) $\langle\langle stml,Sn,sto\rangle,\langle stml,n,STMTL(stml,sto)\rangle\rangle$
(f) $\langle\langle stml,Sn,sto\rangle,\langle stml,sto\rangle\rangle$.

By 6.1(i), (a) - (e) are in $LEX(\supset,O_2(R))$, while (f) $\in LEX(\supset,O_2(R))$ follows from 6.1(ii).

7. Base-total term relations

It is clear from Def. 6.8 that the property of a term relation R to be directly decreasing is decidable. The same holds true for "recursively decreasing": Since the sets of binary relations on OP resp. S are finite, the inductive definition of $REC(O_1,O_2)$ (6.3) induces a recursive algorithm that decides whether R is recursively decreasing. But despite its generality, "recursively decreasing" is not necessary for the normalization property (cf. 4.4).

The matter stands differently with respect to the second condition on R which - together with normalization - implies completeness of PAR, namely base-completeness (cf. 4.4). We shall show in this chapter that base-completeness of R is decidable provided that R is linear (cf. 4.12) and for all $<l,r>\in R$ l contains at least one operation symbol of OP-BOP. The decidablility of base-completeness goes back to the decidability of what is called completeness of a set of term tupels in Huet, Hullot /32/ and what will here be called genericity:

For $w\in BS^*$ a subset M of BT_w is w-generating if each $t\in BT_w$ with $sort(var(t))\subseteq PS$ is a substitution instance of some term tupel in M. For all $\sigma\in$OP-BOP let $arg(R,\sigma)$ be the set of term tupels t such that σt is the lefthand side of some rule in R. R is called base-total if for all $\sigma\in$OP-BOP $arg(R,\sigma)$ is $arity(\sigma)$-generating. It is easy to see that base-totality implies base-completeness. The converse presupposes that R is linear and for all $<l,r>\in R$ $l\notin BT$ (Thm.7.2). Thm. 7.3 shows that the set of w-generating finite term tupel sets is decidable. Hence, base-completeness of linear relations $R\subseteq(T-BT)xT$ is decidable, too (7.5).

7.1 Definition

Let $w \in BS^*$. A set M of linear term tupels $t \in BT_w$ with $sort(op(t)) \cap PS = \emptyset$ is w-generating if for all $u \in BT_w$ with $sort(op(u)) \cap PS = \emptyset$ and $sort(var(u)) \subseteq PS$ there are $t \in M$ and $f \in Z(T)$ with $ft = u$.
$R \subseteq T^2$ is base-total (w.r.t. BOP) if for all $\sigma \in OP-BOP$ with $w = arity(\sigma)$ the set

$$\underline{arg(R,\sigma)} = \{ t \in BT_w \mid t \text{ is linear}, \; sort(op(t)) \cap PS = \emptyset, \; \langle \sigma t, r \rangle \in R \text{ for some } r \}$$

is w-generating.

7.2 Theorem

(i) If R is base-total, then R is base-complete.

(ii) Let $R \subseteq (T-BT) \times T$ be linear and base-complete. For all $A \in K$ and $s \in PS$ let A_s be nonempty. Then R is base-total.

Proof:

(i) Let R be base-total, $A \in K$, $s \in BS$ and $t \in G(A)_s - BG(A)$. R is base-complete if t is not $\xrightarrow[R]{}$-normal. There are $w \in BS^*$ and a subterm $\sigma t'$ of t with $\sigma \in OP-BOP$ and $t' \in BG(A)_w$. Then $ft'' = t'$ for some $t'' \in BT_w$ and $f \in Z(T(A))$ with $sort(op(t'')) \cap PS = \emptyset$ and $sort(var(t'')) \subseteq PS$. By assumption, $arg(R,\sigma)$ is w-generating so that there are $u'' \in arg(R,\sigma)$, $r \in T$ and $g \in Z(T)$ with $\langle \sigma u'', r \rangle \in R$ and $gu'' = t''$. Thus $t \notin NF(R)$ follows from

$$fg\sigma u'' = \sigma fgu'' = \sigma ft'' = \sigma t' \subseteq t.$$

Hence R is base-complete.

(ii) Let $w \in BS^*$, $s \in S$, $\sigma \in OP_{w,s} - BOP$ and $t \in BT_w$

with $sort(op(t)) \cap PS = \emptyset$ and $sort(var(t)) \subseteq PS$.
Let $A \in K$. Since $A_s \neq \emptyset$ for all $s \in PS$, there are $t' \in BG(A)_w$ and $f \in BZ(A)$ such that $ft = t'$.
Base-completeness of R implies $\sigma t' \notin NF(R)$ and thus $\sigma t \notin NF(R)$ because $R \subseteq T^2$.
Since for $<l,r> \in R$ $l \notin BT$, we have $t_i \in NF(R)$ for all $1 \leq i \leq lg(w)$. Hence $fu = t$ for some $u \in arg(R,\sigma)$ and $f \in Z(T)$.
Therefore, $arg(R,\sigma)$ is w-generating, and we conclude that R is base-total. □

7.3 Theorem

There is an algorithm which decides for all $w \in BS^*$ the set of w-generating finite subsets of BT_w.

Proof:

FP(M) denotes the set of all finite subsets of a set M. The algorithm is given by an inductively defined predicate

$$p: \bigcup_{w \in BS^*} FP(BT_w) \longrightarrow \{true, false\}$$

satisfying

$$p(M) = true \text{ iff } M \text{ is w-generating} \qquad (*)$$

for all $M \in FP(BT_w)$ and $w \in BS^*$.
p is defined as follows:

(i) $p(\emptyset) = false$

(ii) $p(\{\varepsilon\}) = true$

(iii) for all $s \in BS$, $w \in BS^*$, $V \in FP(X_s)$ and $M_x \in FP(BT_w)$, $x \in V$,

$$p\left(\bigcup_{x\in V} x\cdot M_x\right) = \begin{cases} p\left(\bigcup_{x\in V} M_x\right) & \text{if for all } x\in V \\ & x\notin \mathrm{var}(M_x) \\ \text{false} & \text{otherwise} \end{cases}$$

(iv) for all $s\in BS$, $w\in BS^*$, $V\in FP(X_s)$, $R\in FP(BT_s - X) - \{\emptyset\}$ and $M_t\in FP(BT_w)$, $t\in V\cup R$,

$$p\left(\bigcup_{t\in V\cup R} t\cdot M_t\right) = \begin{cases} \bigwedge_{\sigma\in BOP_s} p\left(\left(\bigcup_{x\in V} z_{\sigma,x}\cdot M_x\right)\cup\left(\bigcup_{\sigma t\in R} t\cdot M_{\sigma t}\right)\right) \\ \qquad \text{if } s\notin PS \text{ and for all } x\in V \\ \qquad x\notin \mathrm{var}(M_x) \\ \text{false} \qquad \text{otherwise} \end{cases}$$

where $z_{\sigma,x}\in X_{\mathrm{arity}(\sigma)}$ and for all $1\leq i\leq \mathrm{rank}(\sigma)$ $z_{\sigma,x,i}\notin \mathrm{var}(M_x)$.

Let $w\in BS^*$ and $M\in FP(BT_w)$. Existence and uniqueness of $p(M)$ follows from (i) - (iv) by induction on $<\sum_{t\in M} \mathrm{size}(t), \mathrm{lg}(w)>$ with respect to the lexicographic ordering on N^2. Hence (*) holds true if $p': \bigcup_{w\in BS^*} FP(BT_w) \longrightarrow \{\text{true}, \text{false}\}$ such that

$$p'(M) = \begin{cases} \text{true} & \text{if } M \text{ is } w\text{-generating} \\ \text{false} & \text{otherwise} \end{cases}$$

satisfies (i) - (iv) with p' instead of p. Thus we have to show (i') - (iv') below:

(i') For all $w\in BS^*$, $\emptyset$ is not w-generating.

(ii') M is λ-generating iff $M = \{\varepsilon\}$.

(iii') For all $s\in BS$, $w\in BS^*$, $V\in FP(X_s)$ and $M_x\in FP(BT_w)$, $x\in V$, $\bigcup_{x\in V} x\cdot M_x$ is sw-generating iff for all $x\in V$ $x\notin \mathrm{var}(M_x)$ and $\bigcup_{x\in V} M_x$ is w-generating.

(iv') For all $s \in BS$, $w \in BS^*$, $V \in FP(X_s)$, $R \in FP(BT_s - X) - \{\emptyset\}$ and $M_t \in FP(BT_w)$, $t \in V \cup R$,
$\bigcup_{t \in V \cup R} t \cdot M_t$ is sw-generating
iff $s \notin PS$, for all $x \in V$ $x \notin var(M_x)$, and
for all $\sigma \in BOP_s$
$(\bigcup_{x \in V} z_{\sigma,x} \cdot M_x) \cup (\bigcup_{\sigma t \in R} t \cdot M_{\sigma t})$
is arity(σ)w-generating where $z_{\sigma,x} \in X_{arity(\sigma)}$ and for all
$1 \le i \le rank(\sigma)$ $z_{\sigma,x,i} \notin var(M_x)$.

(i') and (ii') are clear.

<u>As to (iii')</u>: Let $s \in BS$, $w \in BS^*$, $V \in FP(X_s)$ and $M_x \in FP(BT_w)$, $x \in V$.

Let $M = \bigcup_{x \in V} x \cdot M_x$ be sw-generating and $u \in BT_w$ such that $sort(op(u)) \cap PS = \emptyset$ and $sort(var(u)) \subseteq PS$. Since all $t \in M$ are linear, we have $x \notin var(M_x)$ for all $x \in V$. Since $BG(A)_s$ is nonempty, there is $u' \in BT_s$ with $sort(op(u')) \cap PS = \emptyset$ and $sort(var(u')) \subseteq PS$. Thus
$f(xt) = u'u$ for some $x \in V$, $t \in M_x$ and $f \in Z(T)$ because M is sw-generating. Hence $ft = u$ so that $\bigcup_{x \in V} M_x$ is w-generating.

Vice versa, suppose that all $x \in V$ satisfy $x \notin var(M_x)$ and $\bigcup_{x \in V} M_x$ is w-generating. Let $u' \in BT_s$ and $u \in BT_w$ with $sort(op(u'u)) = \emptyset$ and $sort(var(u'u)) \subseteq PS$. Then there are $x \in V$, $t \in M_x$ and $f \in Z(T)$ with $ft = u$. Since $x \notin var(t)$,
we may assume w.l.o.g. that
$fx = u'$. Hence $f(xt) = u'u$, and we conclude that
$\bigcup_{x \in V} x \cdot M_x$ is sw-generating.

<u>As to (iv')</u>: Let $s \in BS$, $w \in BS^*$, $V \in FP(X_s)$, $R \in FP(BT_s - X) - \{\emptyset\}$ and $M_t \in FP(BT_w)$, $t \in V \cup R$.
Let $M = \bigcup_{t \in V \cup R} t \cdot M_t$ be sw-generating, $\sigma \in BOP_s$, $u \in BT_{arity(\sigma)}$ and $u' \in BT_w$ with $sort(op(uu')) \cap PS = \emptyset$ and $sort(var(uu')) \subseteq PS$. Since R is nonempty, there is $t \in R$ with $t \cdot M_t \in M$. Since M is sw-generating and $root(t) \in op(t)$, we obtain $s = sort(root(t)) \notin PS$ and

$x \notin var(M_x)$ for all $x \in V$ because all term tupels of M are linear.

Furthermore, there are $t \in V \cup R$, $t' \in M_t$ and $f \in Z(T)$ with $f(tt') = (\sigma u)u'$. If $t \in V$, we choose $z_{\sigma,t} \in X_{arity(\sigma)}$ such that for all $1 \leq i \leq rank(\sigma)$ $z_{\sigma,t,i} \notin M_t$ and define $g \in Z(T)$ by

$$gu = \begin{cases} u_i & \text{if } x = z_{\sigma,t,i} \\ fx & \text{otherwise} \end{cases}$$

Therefore, $g(z_{\sigma,t}t') = uu'$. If $t \in R$, then $ft = \sigma u$ and $R \cap X = \emptyset$ imply $f(t''t') = uu'$ for some $t'' \in BT_{arity(\sigma)}$ with $\sigma t'' = t$.
Hence for all $\sigma \in BOP_s$, $(\bigcup_{x \in V} z_{\sigma,x} \cdot M_x) \cup (\bigcup_{\sigma t \in R} t \cdot M_{\sigma t})$ is arity(σ)w-generating.
Vice versa, suppose that $s \notin PS$ and for all $x \in V$ $x \notin var(M_x)$. For all $\sigma \in BOP_s$ let $M = (\bigcup_{x \in V} z_{\sigma,x} \cdot M_x) \cup (\bigcup_{\sigma t \in R} t \cdot M_t)$ be arity(σ)w-generating where no component of $z_{\sigma,x} \in X_{arity(\sigma)}$ belongs to M_x. Let $u \in BT_s$ and $u' \in BT_w$ such that $sort(op(uu')) \cap PS = \emptyset$ and $sort(var(uu')) \subseteq PS$. Since $s \notin PS$, there are $\sigma \in BOP_s$ and $u'' \in BT_{arity(\sigma)}$ with $\sigma u'' = u$. Since M is arity(σ)w-generating, we obtain $f \in Z(T)$ as well as

(a) $x \in V$ and $t' \in M_x$ with $f(z_{\sigma,x}t') = u''u$

or

(b) $\sigma t \in R$ and $t' \in M_{\sigma t}$ with $f(tt') = u''u$.

In case (a) we define $g \in Z(T)$ by

$$gz = \begin{cases} u & \text{if } z = x \\ fz & \text{otherwise} \end{cases}$$

and obtain $g(xt') = uu'$ because $x \in var(t')$.

Case (b) implies $f((\sigma t)t') = (\sigma u'')u' = uu'$.

Therefore, $\bigcup_{t \in V \cup R} t \cdot M_t$ is sw-generating. □

The stepwise computation of p is illustrated at the following example taken from Huet, Hullot /32/.

7.4 Example

Let PSPEC be empty and BSPEC = nat (cf. 1.3).

Let $M \subseteq BT_{\underline{nat}\ \underline{nat}}$ be defined by

$$M = \{\langle x,0\rangle, \langle 0,Sx\rangle, \langle Sx,S0\rangle, \langle Sx,SSy\rangle\}.$$

Using 7.3 (ii) - (iv) we obtain the following computation of p(M):

$$\begin{aligned}
p(M) &= p(\{0,Sx\}) \cdot p(\{\langle z,0\rangle, \langle x,S0\rangle, \langle x,SSy\rangle\}) \\
&= p(\{\varepsilon\}) \cdot p(\{x\}) \cdot p(\{0,S0,SSy\}) \\
&= p(\{x\}) \cdot p(\{0,S0,SSy\}) \\
&= p(\{\varepsilon\}) \cdot p(\{\varepsilon\}) \cdot p(\{0,Sy\}) \\
&= p(\{0,Sy\}) \\
&= p(\{\varepsilon\})\ p(\{y\}) \\
&= p(\{y\}) \\
&= p(\{\varepsilon\}) \\
&= true.
\end{aligned}$$

Hence 7.3(*) implies that M is $\underline{nat}\ \underline{nat}$-generating.

7.5 Corollary

Let $R \subseteq (T-BT) x T$ be finite and linear, and for all $A \in K$ and $s \in PS$ let A_s be nonempty. Then base-completeness of R is decidable.

Proof:

By assumption and Thm. 7.2, R is base-complete iff R is base-total. By Thm. 7.3, there is an algorithm which decides for all $\sigma \in$ OP-BOP whether the set arg(R,σ) (cf. 7.1) is arity(σ)-generating. Since OP is finite, we can thus decide base-totality of R. □

7.6 Corollary

Let PSPEC be the empty specification, K = $\{\emptyset\}$ and E$\subseteq$(T-BT)xT be finite, linear, normalizing and absolutely confluent. Then completeness of PAR w.r.t. <BPAR,K> is decidable.

Proof:

By assumption and Thm. 4.11, base-completeness of E is sufficient for completeness of PAR.
By assumption and Corollary 7.5, base-completeness of E is decidable. Hence we apply the decision procedure to E. If it returns an affirmative answer, PAR is complete. If it stops with a negative answer, then E is not base-complete, i.e. some $\xrightarrow[E]{}$-normal form t$\in$G does not belong to BG. If PAR would be complete w.r.t. <BPAR,K>, some t'$\in$BG would satisfy $t \equiv_{SPEC} t'$. Absolute confluence of E would imply $t \xrightarrow[E]{*} \xleftarrow[E]{*} t'$. Since t$\in$NF(E), we would get $t \xleftarrow[E]{*} t'$. Since t'$\in$BG and E$\subseteq$(T-BT)xT, we would have to conclude t = t' which contradicts t$\notin$BG.
Hence PAR is not complete if the decision procedure for base-completeness returns a negative answer. □

Although we achieved it differently, Corollary 7.6 just states the main result of Nipkow, Weikum /49/ (Thm. 3.1).
Thms. 7.2(i) and 6.11 yield

7.7 Completeness Theorem

Let $H \subseteq T^2$ be a term rewriting system such that for all $\langle l,r\rangle \in H$, $A \in K$ and $f \in BZ(G(A))$ $fl \equiv_{SPEC(A)} fr$. If some directly decreasing and base-total relation R satisfies $H \subseteq R \subseteq H \cup E$, then PAR is complete w.r.t. $\langle BPAR, K\rangle$. $\square$

We thus obtained the completeness theorem which provides the foundation for the first part of our decision graph for completeness given in 3.1. We have seen that both requirements to R occuring in that graph are decidable. Two examples described in the following may illustrate the usefulness of Thm. 7.7. Further specifications treated by a variant of Thm. 7.7 will be given in the next chapter.

7.8 Example (Ackermann function)

Let BSPEC = nat and SPEC be as in Example 6.9. There we have shown that R = {e1, e2, e3} is directly decreasing. Let p be the predicate defined in Thm. 7.3. We get

$$\begin{aligned} & p(arg(R,ACK)) = p(\{\langle 0,x\rangle,\langle Sx,0\rangle,\langle Sx,Sy\rangle\}) \\ & = p(\{x\})\cdot p(\{\langle x,0\rangle,\langle x,Sy\rangle\}) = p(\{\varepsilon\})\cdot p(\{0,Sy\}) \\ & = p(\{0,Sy\}) = p(\{\varepsilon\})\cdot p(\{y\}) = p(\{y\}) = p(\{\varepsilon\}) \\ & = true. \end{aligned}$$

Hence by 7.3 (*), arg(R,ACK) is nat nat-generating, and thus R is base-total.
Therefore, Thm. 7.7 implies that PAR is complete w.r.t. $\langle BPAR, K\rangle$.

7.9 Example (int)

Let PSPEC be empty,

BSPEC = int

sorts:	int	
opns:	$0: \longrightarrow$ int	
	S,P: int int $\longrightarrow$ int	
eqns:	SPx = x	i1
	PSx = x	i2

and

SPEC = int +

opns:	$+,-,\cdot$: int int $\longrightarrow$ int	
eqns:	$x + 0 = x$	i3
	$x + Sy = S(x+y)$	i4
	$x + Py = P(x+y)$	i5
	$x - 0 = x$	i6
	$x - Sy = P(x-y)$	i7
	$x - Py = S(x-y)$	i8
	$x \cdot 0 = 0$	i9
	$x \cdot Sy = (x \cdot y)+x$	i10
	$x \cdot Py = (x \cdot y)-x$	i11

Then

$$O_1(E) = \{<+,S>,<+,+>,<+,P>,<-,P>,<-,->,<-,S>, <\cdot,0>,<\cdot,\cdot>,<\cdot,+>,<\cdot,->\}$$

(cf. 6.8).

Choosing $O_2(E) = \emptyset$, E is directly decreasing:
The righthand sides of i1, i2, i3 and i6 do not contain operation symbols. The only subterm in the righthand sides of i4 and i5 with $<root(t),root(l)> \in O_1(E)^*$ is x+y. Since $<<x,Sy>,<x,y>>$ as well as $<<x,Py>,<x,y>>$ belong to $LEX(\supset,\emptyset)$, 6.8(*) holds true for $<l,r> \in \{i4,i5\}$ and $t = x+y$. Accordingly, i7, i8, i10 and i11 have to be checked.

Let p be the predicate defined in Thm. 7.3. We get for all $\sigma \in OP\text{-}BOP = \{+,-,\cdot\}$

$$p(arg(E,\sigma)) = p(\{<x,0>,<x,Sy>,<x,Py>\})$$
$$= p(\{0,Sy,Py\}) = p(\{\varepsilon\}) \cdot p(\{y\}) \cdot p(\{y\})$$
$$= p(\{y\}) = p(\{\varepsilon\}) = \text{true}.$$

Hence by 7.3(*), for all $\sigma \in OP\text{-}BOP$ arg(E,σ) is int int-generating, and thus E is base-total. Therefore, Thm. 7.7 implies that PAR is complete w.r.t. <BPAR,K>.

8. Complete specifications with conditionals

Now we treat the case where for all $A \in K$ BSPEC(A) is a correct extension of bool (cf. 1.2 and 2.5; since bool is a non-parameterized specification, "correct extension of bool" means of course "correct extension of $\langle\langle\emptyset,\emptyset,\emptyset\rangle,\text{bool}\rangle,\{\emptyset\}\rangle$".) This case is characterized as follows:

8.1 Proposition

Let $A \in K$ and bool be a subspecifiation of BSPEC(A). BSPEC(A) is a correct extension of bool iff

(*) for all $t \in BG(A)_{\underline{bool}}$ either $t \equiv_{BSPEC(A)} TRUE$ or $t \equiv_{BSPEC(A)} FALSE$.

Proof:

Let $bool0 = \langle\{\underline{bool}\},\{TRUE,FALSE:\longrightarrow \underline{bool}\},\emptyset\rangle$ and R be the set of equations of bool. By Thm. 2.8, (*) holds true iff BSPEC(A) is a correct extension of bool0. Let BSPEC(A) be a correct extension of bool. In order to conclude (*) we have to show that bool is a correct extension of bool0.

Clearly, R is directly decreasing and base-total. Hence Thm. 7.7 implies that bool is complete w.r.t. bool0. Since the lefthand sides of each two different equations of bool do not overlap, R is absolutely confluent by Prop. 4.13. Since for all $\langle l,r\rangle \in R$ $op(l) \cap \{TRUE,FALSE\} = \emptyset$, R is base-consistent w.r.t. bool0 by Lemma 4.18. Thus Thm. 4.19 implies that bool is consistent w.r.t. bool0. Therefore, bool is a correct extension of bool0 by Thm. 2.8.

Vice versa, suppose that (*) holds true, i.e. BSPEC(A) is a correct extension of bool0. Clearly, BSPEC(A) is complete w.r.t. bool. Let t,t' be two terms that consist of operation symbols of bool and satisfy $t \equiv_{BSPEC(A)} t$. Since $TRUE \not\equiv_{BSPEC(A)} FALSE$ and BSPEC(A) contains bool, we obtain

$$t \equiv_{bool} TRUE \equiv_{bool} t' \text{ or } t \equiv_{bool} FALSE \equiv_{bool} t'$$

Hence BSPEC(A) is consistent w.r.t. bool so that by Thm. 2.8, BSPEC(A) is a correct extension of bool. □

<u>For the rest of this chapter we assume that for all A∈K BSPEC(A) is a correct extension of bool.</u>

8.2 <u>Definition</u>

Let s∈BS. $IF \in BOP_{\underline{bool}\ ss,s}$ is a <u>conditional</u> if two equations

$$IF(TRUE,x,y) = x \text{ and } IF(FALSE,x,y) = y,$$

called <u>conditional rules</u>, with $x,y \in X, x \neq y$, belong to BE. <u>CR</u> denotes the set of all conditional rules.

We fix a set C of conditionals.

8.3 <u>Definition</u>

The term relation <u>CCR</u> of <u>conditional-compatibility rules</u> consists of all linear rules

$$<\sigma(v,IF(b,x,y),w),\ IF'(b,\sigma(v,x,w),\sigma(v,y,w))>$$

with $\sigma \in OP-(C \cup POP)$, $IF,IF' \in C$, $b,x,y \in X$ and $v,w \in X^*$.

8.4 <u>Definition</u> (cf. 7.1)

Let $\overline{BOP}$ be a subset of BOP and $\overline{BT} = T_{\overline{BOP}}$. Let

$w \in BS^*$. A set M of linear term tupels $t \in BT_w$ with $sort(op(t)) \cap PS = \emptyset$ is <u>w-generating w.r.t. $\overline{BOP}$</u> if for all $u \in \overline{BT}_w$ with $sort(op(u)) \cap PS = \emptyset$ and $sort(var(u)) \subseteq PS$ there are $t \in M$ and $f \in Z(T)$ with $ft = u$.

$R \subseteq T^2$ is <u>base-total w.r.t. $\overline{BOP}$</u> if for all $\sigma \in OP-BOP$ the set $arg(R,\sigma)$ (cf.7.1) is $arity(\sigma)$-generating w.r.t. $\overline{BOP}$.

8.5 <u>Completeness Theorem</u>

If some directly decreasing and w.r.t. BOP-C base-total relation $R \subseteq E$ satisfies $root(l) \notin C$ for all $\langle l,r \rangle \in R$, then PAR is complete w.r.t. $\langle BPAR,K \rangle$.

<u>Proof:</u>

Let $\overline{BOP} = BOP-C$, $\overline{BT} = T_{\overline{BOP}}$ and $\overline{R} = R \cup CCR$. By Thm. 7.7, PAR is complete w.r.t. $\langle BPAR,K \rangle$ if

(a) for all $\langle l,r \rangle \in CCR$, $A \in K$ and $f \in BZ(G(A))$
$fl \equiv_{SPEC(A)} fr$,

(b) $\overline{R}$ is directly decreasing and base-total.

Let $\langle l,r \rangle \in CCR$. Then there are $\sigma \in OP-C$, $IF, IF' \in C$, $b,x,y \in X$ and $v,w \in X^*$ such that $l = \sigma(v, IF(b,x,y), w)$ and
$r = IF'(b, \sigma(v,x,w), \sigma(v,y,w))$. Let $A \in K$ and $f \in BZ(G(A))$.
By Lemma 8.1, $fb \equiv_{BSPEC(A)} TRUE$ or
$fb \equiv_{BSPEC(A)} FALSE$.
Hence w.l.o.g.

$$\begin{aligned} fl &\equiv_{SPEC(A)} \sigma(fv, IF(TRUE,fx,fy), fw) \\ &\equiv_{SPEC(A)} \sigma(fv,fx,fw) \\ &\equiv_{SPEC(A)} IF'(TRUE, \sigma(fv,fx,fw), \sigma(fv,fy,fw)) \\ &\equiv_{SPEC(A)} fr. \end{aligned}$$

Thus (a) holds true.

Concerning (b) note that by assumption, $\bar{R}$ is directly decreasing if for all $<l,r>\in\bar{R}$ and all subterms t of r, $<root(t),root(l)>\in O_1(\bar{R})^*$ implies $<root(t),root(l)>\in O_1(R)^*$. So let $<l,r>\in\bar{R}$, and $t\leq r$ with $<root(t),root(l)>\in O_1(R)^*$. If $<root(t),root(l)>\notin O_1(R)^*$, there would be $<l',r'>\in CCR$ such that $<root(t),root(l')>\in O_1(\bar{R})^*$ and $<IF,root(l)>\in O_1(R)^*$ for some $IF\in op(r')$. But then either $IF = root(l)$ or $IF = root(l")$ for some $<l",r">\in R$.
Both cases contradict the fact that for all $<l,r>\in\bar{R}$ $root(l)\notin C$.

Hence $\bar{R}$ is directly decreasing.
Let $A\in K$, $s\in BS$ and $t\in G(A)_s-BG(A)$. Then there are $\sigma\in OP-BOP$ and $t'\in BG(A)^*$ such that $\sigma t'$ is a subterm of t. Thus $ft" = t'$ for some $t"\in BT^*$ and $f\in Z(T(A))$ with $sort(op(t"))\cap PS = \emptyset$ and $sort(var(t"))\subseteq PS$. If $t"$ does not contain conditionals, then by base-totality of R w.r.t. $\overline{BOP}$, $gu" = t"$ for some $u"\in arg(R,\sigma)$ and $f\in Z(T)$. Since $fg\sigma u" = \sigma fgu" = \sigma ft" = \sigma t'\leq t$, t is not $\xrightarrow[R]{}$-normal. If $t"$ contains a conditional, then $\sigma t"\xrightarrow[CCR]{} t_o$ for some t_o so that again $t\notin NF(R)$.
Therefore, $\bar{R}$ is base-total. $\square$

8.6 <u>Example</u> (arrayl)

Let BPAR = <entry,array> (cf. 1.5), PAR = <entry,arrayl> (cf. 2.2), K be the class of entry-algebras defined in 1.11 and $C = \{IFE,IFA\}$.
Clearly, BPAR is a correct extension of <BPAR,K>.
Hence by Thm. 2.17, BPAR is persistent w.r.t. K because

$$sort(BOP-POP) = sort(BE-PE) = \emptyset.$$

Thus Prop. 2.16 implies that BPAR is a correct extension of <<entry,entry>, K>. By Thm. 2.7, BPAR is complete and consistent w.r.t. <<entry,entry>,K>, so that by Prop 2.19,

(i) for all $A \in K$ and $t \in BG(A)_{\underline{bool}}$ $t \equiv_{BSPEC(A)} a$ for some $a \in A_{\underline{bool}}$

and

(ii) for all $A \in K$ and $a,a' \in A_{\underline{bool}}$ $a \equiv_{BSPEC(A)} a'$ implies $a = a'$.

Since $A_{\underline{bool}} = \{TRUE_A, FALSE_A\}$, 8.1(*) follows from (i) and (ii), and thus by Lemma 8.1, for all $A \in K$ BSPEC(A) is a correct extension of bool.

Now completeness of PAR w.r.t. < BPAR,K > is shown using Thm. 8.5: Let $R = \{\underline{a5}, \underline{a6}\}$ (cf. 2.2). We obtain

$$O_1(R) = \{<GET,\sigma> / \sigma \in \{UNDEF, IFE, EQN, GET\}\}$$

and choose $O_2(R) = \emptyset$. The righthand side of a5 has no subterms t with $<root(t),GET> \in O_1(R)^*$. The only subterm t of the righthand side of a6 with this property is GET(a,m). Since

$$<<PUT(a,n,x),m>,<a,m>> \in LEX(\supset,\emptyset),$$

condition 6.8(*) holds true for $<l,r> = \underline{a6}$. Hence R is directly decreasing.

Base-totality w.r.t. BOP-C can be decided like base-totality w.r.t. BOP (cf. chapter 7): Replacing BOP by BOP-C and BT by T_{BOP-C} in 7.3(iii)-(iv), the predicate p in Thm. 7.3 induces a decision algorithm for subsets of $T_{BOP-C,w'}$, $w \in BS^*$, which are w-generating w.r.t. BOP-C. We get

$$p(arg(R,GET)) = p(\{<NEW,n>,<PUT(a,n,x),m>\})$$
$$= p(\{n\})\cdot p(\{<a,n,x,m>\}) = p(\{\varepsilon\})\cdot p(\{<n,x,m>\})$$
$$= p(\{<x,m>\}) = p(\{m\}) = p(\{\varepsilon\}) = true.$$

Hence R is base-total w.r.t. BOP-C.
Therefore, Thm. 8.5 implies that PAR is complete w.r.t. <BPAR,K>. □

8.7 Example (setl)

Let entry be the parameter specification given in Ex. 1.5,

```
  set = entry +
      sorts: set
      opns:  Ø: ⟶ set
             INS: set entry ⟶ set
             IFS: bool set set ⟶ set
      eqns:  INS(INS(s,x),x) = INS(s,x)                s1
             INS(INS(s,x),y) = INS(INS(s,y),x)         s2
             IFS(TRUE,s,s') = s                        s3
             IFS(FALSE,s,s') = s'                      s4
```

and

```
  setl = set +
      opns:  DEL: set entry ⟶ set
      eqns:  DEL(Ø,x) = Ø                              s5
             DEL(INS(s,x),y) = IFS(EQE(x,y),
                                   DEL(s,y),
                                   INS(DEL(s,y),x))
                                                       s6
```

Let BPAR = <entry,set>, PAR = <entry,setl>, K be the class of entry-algebras defined in 1.11 and

$C = \{IFS\}$. The proof that for all $A \in K$ BSPEC(A) is a correct extension of bool and that PAR is complete w.r.t. $<BPAR,K>$ proceeds analogously to Ex. 8.6 and is left to the reader. □

The existence of a directly decreasing and base-total relation $R \subseteq E$ depends upon the syntax of SPEC. In the following we develop a completeness criterion which takes into account the semantics of BPAR, particularly our assumption that for all $A \in K$ BSPEC(A) is a correct extension of bool. The terms of BT_{bool} will be called <u>base predicates</u>. The set of base predicates is denoted by <u>BP</u>.

8.8 <u>Definition</u>

A term <u>t</u> is <u>simple</u> if $root(t) \in \bigcup_{s \in BS} OP_s - BOP$ and $op(arg(t)) \subseteq BOP \cup \bigcup_{s \in S-BS} OP_s$.
<u>ST</u> denotes the set of all $t \in \bigcup_{s \in BS} T_s$ such that all subterms t' of t with $root(t) \in OP-BOP$ are simple.
$\underline{R \subseteq T^2}$ is <u>simple</u> if for all $<l,r> \in R$ l is simple and $r \in ST$.

Next we define the set of conditional subterms of some $t \in T$ which are just those pairs $<t',p>$ where $t' \leq t$ and p is the conjunction of all base predicates occurring as conditions of conditionals preciding t' . Formally we have

8.9 <u>Definition</u>

The sets <u>sct(t,p)</u>, $t \in T$, $p \in BP$, of <u>simple conditional subterms of t w.r.t. p</u> are inductively defined as follows:

$$sct(t,p) = \begin{cases} \{<t,p>\} & \text{if } t \text{ is simple} \\ \{<t',q\wedge q'>/<t',q>\in sct(u,p)\} \cup \\ \{<t',q\wedge\neg q'>/<t',q>\in sct(u',p)\} & \text{if } t = IF(q',u,u'), \\ & IF\in C \text{ and } q'\in BP \\ \bigcup_{i=1}^{lg(arg(t))} sct(arg(t)_i,p) & \text{otherwise} \end{cases}$$

8.10 <u>Definition</u>

A simple term relation R is called <u>conditionally decreasing</u> if for all $A\in K$ and $\sigma\in OP-BOP$ there is a weight function

$$w_{\sigma,A}: BG(A)_{arity(\sigma)} \longrightarrow \mathbb{N}$$

such that for all $<l,r>\in R$, $f\in BZ(G(A))$ and $<t,p>\in sct(r,TRUE)$

(*) $fp \equiv_{BSPEC(A)} TRUE$ implies
$w_{root(t),A}(t') < w_{root(l),A}(f\circ arg(l))$
for some $t'\in BG(A)^*$ with $t' \equiv_{BSPEC(A)} f\circ arg(t)$.

Property (*) will be abbreviated by <u>dec(t,p)</u>.

Let $A\in K$. For proof purposes we define for all simple terms $l\in T$ and $f\in BZ(G(A))$ a predicate

$$\underline{pred}_{l,f}: ST \times BP \longrightarrow \{true, false\}$$

by:

$pred_{l,f}(t,p) = true$ iff for all $<t',p'>\in sct(t,p)$ $dec(t',p')$ holds.

8.11 <u>Proposition</u>

Let $l \in T$ be simple and $f \in BZ(G(A))$. $pred_{l,f}$ satisfies the following recursive definition:

$$pred_{l,f}(t,p) = \begin{cases} \text{true} & \text{if } t \text{ is simple and } dec(t,p) \text{ holds} \\ \text{false} & \text{if } t \text{ is simple and } dec(t,p) \text{ does not hold} \\ pred_{l,f}(u,p\wedge q)\cdot pred_{l,f}(u',p\wedge\neg q) & \text{if } t=IF(q,u,u'),\ IF\in C \text{ and } q\in BP \\ \bigwedge_{i=1}^{lg(arg(t))} pred_{l,f}(arg(t)_i,p) & \text{otherwise} \end{cases}$$

<u>Proof:</u>

If t is simple, then $sct(t,p) = \{<t,p>\}$. Hence by definition of $pred_{l,f}$, if $dec(t,p)$ holds, then $pred_{l,f}(t,p)$ = true, otherwise $pred_{l,f}(t,p)$ = false.

Let $u = IF(q,u,u')$, $IF \in C$ and $q \in BP$. We have to show that

(a) for all $<t',p'> \in sct(t,p)$ $dec(t',p')$ holds

iff

(b) for all $<t',p'> \in sct(u,p\wedge q)$ and $<t'',p''> \in sct(u',p\wedge\neg q)$ $dec(t',p')$ and $dec(t'',p'')$ hold.

So assume (a) and let $<t',p'> \in sct(u,p\wedge q)$ and $<t'',p''> \in sct(u',p\wedge\neg q)$. By Def. 8.9,

$<t',p'\wedge q>\in sct(t,p\,q)$ and $<t'',p''\wedge\neg q>\in sct(t,p\wedge\neg q)$. Hence $<t',p'>,<t'',p''>\in sct(t,p)$ so that by assumption, $dec(t',p')$ and $dec(t'',p'')$ hold.
Vice versa, assume (b) and let $<t',p'>\in sct(t,p)$. Then there is $q'\in BP$ such that w.l.o.g. $p'=q'\wedge q$ and $<t',q'>\in sct(u,p)$. Thus $<t',p'>\in sct(u,p\wedge q)$ and by assumption, $dec(t',p')$ holds.
Let $\sigma\in BOP$ and $u\in T^*$ with $\sigma u = t$. We have to show that (a) above holds iff for all $1\leq i\leq lg(u)$ and $<t',p'>\in sct(u_i,p)$ $dec(t',p')$ is satisfied. But this follows immediately from Def. 8.9 which implies

$$sct(t,p) = \bigcup_{i=1}^{lg(u)} sct(u_i,p). \square$$

Provided that for all simple terms t and all base predicates p we can decide dec(t,p), Prop. 8.11 induces an algorithm for deciding whether a simple term relation is conditionally decreasing:

8.12 Proposition

A simple term relation R is conditionally decreasing if for all $A\in K$ and $\sigma\in OP-BOP$ there is a weight function

$$w_{\sigma,A}: BG(A)_{arity(\sigma)} \longrightarrow \mathbb{N}$$

such that for all $<l,r>\in R$ and $f\in BZ(G(A))$ $pred_{l,f}(r,TRUE) = true$. $\square$

An example of a conditionally decreasing relation was already given in chapter 3. A more complicated example will be described in the following.It is a variant of the parts-system specification presented in Ehrig, Fey /17/. Example 8.13 below contains the base specification BPAR as well as a correctness proof of BPAR that uses the correctness criterion 1.15 analogously to the correctness proof of array given in Ex. 1.16. In Example 8.14 we shall extend BPAR to PAR and define appropriate weight functions in order to show that R = E-BE is conditionally decreasing. Lemma 8.14A states a property of these weight functions which will be needed in the proof that E-BE is conditionally decreasing (Ex. 8.15).

8.13 Example (parts-system)

Let entry be the parameter specification given in Ex. 1.5,

```
counted-set = entry +
      sorts: bag
      opns:  Ø:⟶ bag
             INS: bag entry nat ⟶ bag
             IFC: bool bag bag ⟶ bag
      eqns:  INS(INS(b,x,n),x,m) =
                     INS(b,x,n+m)
             INS(INS(b,x,n),y,m) =
                     INS(INS(b,y,m),x,n)
             IFC(TRUE,b,b') = b
             IFC(FALSE,b,b') = b'

counted-set1 = counted-set +
      opns:  JOIN: bag bag ⟶ bag
             DEL: bag entry ⟶ bag
      eqns:  JOIN(Ø,b) = b
             JOIN(INS(b,x,n),b') =
                     INS(JOIN(b,b'),x,n)
             DEL(Ø,x) = Ø
             DEL(INS(b,x,n),y) =
                     IFB(EQE(x,y),
                         DEL(b,y),
                         INS(DEL(b,y),x,n))
```

```
partssystem = counted-setl +
       sorts:  graph
       opns:   EMPTY:⟶graph
               ADD-PART: graph entry⟶graph
               ADD-LINK: graph entry nat entry    graph
               IS-PART: graph entry⟶bool
               IS-LINK: graph entry entry ⟶bool
               LINK-ERROR: graph entry nat entry⟶bool
               IFG: bool graph graph ⟶graph
       eqns:   ADD-PART(ADD-PART(g,x),x) = ADD-PART(g,x)
               ADD-PART(ADD-PART(g,x),y)
                 = ADD-PART(ADD-PART(g,y),x)
               ADD-PART(ADD-LINK(g,x,n,y),z)
                 = IFG(LINK-ERROR(g,x,n,y),
                       ADD-PART(g,z),
                       IFG(EQE(x,z)
                           ∨EQE(y,z),
                           ADD-LINK(g,x,n,y),
                           ADD-LINK(ADD-PART(g,z),x,n,y)))
               ADD-LINK(EMPTY,x,n,y) = EMPTY
               ADD-LINK(ADD-PART(g,x),y,n,z)
                 = IFG(LINK-ERROR(ADD-PART(g,x),y,n,z),
                       ADD-PART(g,x),
                       IFG(EQE(x,y)∨EQE(x,z),
                           ADD-LINK(ADD-PART(g,x),y,n,z),
                           ADD-PART(ADD-LINK(g,y,n,z),x)))
               ADD-LINK(ADD-LINK(g,x,n,y),y',m,z)
                 = IFG(LINK-ERROR(ADD-LINK(g,x,n,y),y',m,z),
                       ADD-LINK(g,x,n,y),
                       ADD-LINK(ADD-LINK(g,y',m,z),x,n,y))
               LINK-ERROR(g,x,n,y) = ¬IS-PART(g,x)
                                     ∨¬IS-PART(g,y)
                                     ∨EQN(n,0)
                                     ∨IS-LINK(g,y,x)
```

```
IS-PART(EMPTY,x) = FALSE
IS-PART(ADD-PART(g,x),y) = EQE(x,y)∨IS-PART(g,y)
IS-PART(ADD-LINK(g,x,n,y),z) = IS-PART(g,z)
IS-LINK(EMPTY,x,y) = EQE(x,y)
IS-LINK(ADD-PART(g,x),y,z) = IS-LINK(g,y,z)
IS-LINK(ADD-LINK(g,x,n,y),y',z)
  = IFB(LINK-ERROR(g,x,n,y),
        IS-LINK(g,y',z),
        IS-LINK(g,y',z)
          ∨(IS-LINK(g,y',x)∧IS-LINK(g,y,z))
IFG(TRUE,g,g') = g
IFG(FALSE,g,g') = g'
```

Let K be the class of all entry-algebras A which satisfy 1.11 (i), (ii) and

$$EQE_A(a,b) = TRUE_A \text{ iff } a = b.$$

Let BPAR = <entry,parts-system> and

$BF: K \longrightarrow Alg(BSPEC)$

be defined by:

$(BFA)_s = A_s$ for all $s \in PS$,

$(BFA)_{\underline{bag}} = G_{counted\text{-}setl(A),\ \underline{bag}}$'

$(BFA)_{\underline{graph}}$ = the set of all finite, acyclic and directed graphs with nodes in $A_{\underline{entry}}$ and edge colors in $A_{\underline{nat}} - \{0_A\}$,

$\sigma_{BFA} = \sigma_A$ for all $\sigma \in POP$,

$\sigma_{BFA} = \sigma_G$ for all $\sigma \in \{\emptyset, INS, JOIN, DEL\}$ where $G = G_{counted\text{-}setl(A)}$'

$EMPTY_{BFA}$ is the empty graph,

$ADD\text{-}PART_{BFA}(g,e)$ agrees with g if e is not a node of g, otherwise e is added to g,

$ADD\text{-}LINK_{BFA}(g,e,n,e')$ agrees with g if $LINK\text{-}ERROR_{BFA}(g,e,n,e') = TRUE_A$, otherwise an edge from e to e' with color n is added to g,

$IS\text{-}PART_{BFA}(g,e) = TRUE_A$ iff e is a node of g,

$IS\text{-}LINK_{BFA}(g,e,e') = TRUE_A$ iff e = e' or g contains a path from e to e',

$LINK\text{-}ERROR_{BFA}(g,e,n,e') = TRUE_A$ iff e or e' is not a node of g or n = 0 or e = e' or there is a path from e' to e in g.

$IFG_{BFA}(p,g,g') = if(p=TRUE_A)$ then g else g'.

Using Thm. 1.15 we show that <u><BPAR,K> is correct w.r.t. BF</u>:
Since for all $A \in K$ BFA is a BSPEC(A)-algebra with $a_{BFA} = a$ for all $a \in A$, 1.15(i) holds true. Let $A \in K$ and f be an arbitrary function, which maps each $g \in (BFA)_{\underline{graph}}$ to some isolated node of g or to some triple $\langle e,n,e' \rangle \in A_{\underline{entry}} \times A_{\underline{nat}} \times A_{\underline{entry}}$ such that

in the second case g contains an edge from e to e' colored by n. For all $s \in PS$ set $h_s = inc$, let $h_{\underline{bag}}:(BFA)_{\underline{bag}} \longrightarrow BG(A)_{\underline{bag}}$ assign each equivalence class a to some $t \in a$, and define $h = h_{\underline{graph}}:(BFA)_{\underline{graph}} \longrightarrow BG(A)_{\underline{graph}}$ by:

(i) $h(EMPTY_{BFA}) = EMPTY$,

(ii) $h(ADD\text{-}PART_{BFA}(g,e)) = ADD\text{-}PART(hg,e)$
where $e = f(ADD\text{-}PART_{BFA}(g,e))$ does not belong to g,

(iii) $h(ADD\text{-}LINK_{BFA}(g,e,n,e')) = ADD\text{-}LINK(hg,e,n,e')$
where $f(ADD\text{-}LINK_{BFA}(g,e,n,e'))$
$= \langle e,n,e' \rangle$ and $LINK\text{-}ERROR_{BFA}(g,e,n,e')$
$= FALSE_A$.

Clearly, $h_{\underline{bag}}$ is a right-inverse of $eval_{BFA,\underline{bag}}$. The corresponding property for $h_{\underline{graph}}$ will be proved by induction on the number $\underline{\#(g)}$ of nodes and edges of g:
If g is empty, then $eval_{BFA} \circ h(g) = g$ follows from (i). Otherwise there is $g' \in (BFA)_{\underline{graph}}$ such that either $g = ADD\text{-}PART_{BFA}(g',fg)$ and fg does not belong to g' or $g = ADD\text{-}LINK_{BFA}(g',e,n,e')$, fg = $= \langle e,n,e' \rangle$ and g has more edges from e to e' colored by n than g'(and thus by definition of $ADD\text{-}LINK_{BFA}$, $LINK\text{-}ERROR_{BFA}(g,e,n,e') = FALSE_A$). Both cases imply $\#(g') < \#(g)$ so that by induction hypothesis and (ii) resp. (iii), $eval_{BFA} h(g) = g$.

By Thm. 1.15 it remains to show that for all $w \in BS^*$, $s \in BS$, $\sigma \in BOP_{w,s}$-POP and $a \in (BFA)_w$

$$\sigma h(a) \equiv_{BSPEC(A)} h\sigma_{BFA}(a). \qquad (*)$$

For all $\sigma \in \{\emptyset, INS, JOIN, DEL\}$ (*) follows from the definitions of $h_{\underline{bag}}$ and σ_{BFA}. For σ = EMPTY (*) follows from (i). (*) for σ = IFG is an immediate consequence of

$$IFG(TRUE_A, ha, ha') \equiv_{BSPEC(A)} ha$$

and

$$IFG(FALSE_A, ha, ha') \equiv_{BSPEC(A)} ha' .$$

For $\sigma \in \{$ADD-PART, ADD-LINK, IS-PART, IS-LINK, LINK-ERROR$\}$ we show (*) by induction on #(g) ($\equiv$ stands for $\equiv_{BSPEC(A)}$):

Let g = $EMPTY_{BFA}$. Then for all $e, e' \in A_{\underline{entry}}$ and $n \in A_{\underline{nat}}$

$$\begin{aligned}
&\text{ADD-PART}(hg,e) = \text{ADD-PART}(hg, f(\text{ADD-PART}_{BFA}(g,e)))\\
&\qquad\qquad = h(\text{ADD-PART}_{BFA}(g,e)),\\
&\text{ADD-LINK}(hg,e,n,e') = \text{ADD-LINK}(\text{EMPTY},e,n,e')\\
&\qquad\qquad \equiv \text{EMPTY} = hg = h(\text{ADD-LINK}_{BFA}(g,e,n,e')),\\
&\text{IS-PART}(hg,e) = \text{IS-PART}(\text{EMPTY},e) \equiv \text{FALSE}\\
&\qquad \equiv \text{FALSE}_A = \text{IS-PART}_{BFA}(g,e),\\
&\text{IS-LINK}(hg,e,e') = \text{IS-LINK}(\text{EMPTY},e,e')\\
&\qquad \equiv \text{EQE}(e,e') \equiv \text{EQE}_A(e,e')\\
&\qquad = \text{IS-LINK}_{BFA}(g,e,e'),\\
&\text{LINK-ERROR}(hg,e,n,e') \equiv \neg\text{IS-PART}(hg,e)\\
&\qquad \vee \neg\,\text{IS-PART}(hg,e') \vee \text{EQN}(n,0) \vee \text{IS-LINK}(hg,e',n,e)\\
&\qquad \equiv \neg\text{IS-PART}_{BFA}(g,e)\\
&\qquad\quad \vee \neg\,\text{IS-PART}_{BFA}(g,e') \vee \text{EQN}_A(n,0)\\
&\qquad\quad \vee \text{IS-LINK}_{BFA}(g,e',n,e)\\
&\qquad \equiv \neg_A\text{IS-PART}_{BFA}(g,e) \vee_A \neg_A\text{IS-PART}_{BFA}(g,e')\\
&\qquad\quad \vee_A \text{EQN}_A(n,0) \vee_A \text{IS-LINK}_{BFA}(g,e',n,e)\\
&\qquad = \text{LINK-ERROR}_{bfa}(g,e,n,e').
\end{aligned}$$

Let g $\neq$ $EMPTY_{BFA}$, $e, e' \in A_{\underline{entry}}$ and $n \in A_{\underline{nat}}$.

Case 1: There are g_o, e_o with $\text{ADD-PART}_{BFA}(g_o,e_o) = g$ and $\#(g) = \#(g_o)+1$. By induction hypothesis we obtain

$$\begin{aligned}
\text{IS-PART}(hg,e) &\equiv \text{IS-PART}(\text{ADD-PART}(hg_o,e_o),e)\\
&\equiv \text{EQE}(e_o,e) \vee \text{IS-PART}(hg_o,e)\\
&\equiv \text{EQE}_A(e_o,e) \vee_A \text{IS-PART}_{BFA}(g_o,e)\\
&= \text{IS-PART}_{BFA}(g,e),
\end{aligned}$$

$$\begin{aligned}
\text{IS-LINK}(hg,e,n,e') &\equiv \text{IS-LINK}(\text{ADD-PART}(hg_o,e_o),e,n,e')\\
&\equiv \text{IS-LINK}(hg_o,e,n,e') \equiv \text{IS-LINK}_{BFA}(g_o,e,n,e')\\
&= \text{IS-LINK}_{BFA}(g,e,n,e').
\end{aligned}$$

As in the case "$g = \text{EMPTY}_{BFA}$" we get

$$\text{LINK-ERROR}(hg,e,n,e') \equiv \text{LINK-ERROR}_{BFA}(g,e,n,e').$$

Case 2: There are g_o,e_o,n_o,e_o' with $\text{ADD-LINK}_{BFA}(g_o,e_o,n_o,e_o') = g$ and $\#(g) = \#(g_o)+1$. Then by induction hypothesis,

$$\begin{aligned}
\text{IS-PART}(hg,e) &\equiv \text{IS-PART}(\text{ADD-LINK}(hg_o,e_o,n_o,e_o'),e)\\
&\equiv \text{IS-PART}(hg_o,e) \equiv \text{IS-PART}_{BFA}(g_o,e)\\
&= \text{IS-PART}_{BFA}(g,e),
\end{aligned}$$

$$\begin{aligned}
\text{IS-LINK}(hg,e,e') &\equiv \text{IS-LINK}(\text{ADD-LINK}(hg_o,e_o,n_o,e_o'),e,e')\\
&\equiv \text{IFB}(\text{LINK-ERROR}(hg_o,e_o,n_o,e_o'),\\
&\qquad \text{IS-LINK}(hg_o,e,e'),\\
&\qquad \text{IS-LINK}(hg_o,e,e')\\
&\qquad\quad \vee(\text{IS-LINK}(hg_o,e,e_o)\\
&\qquad\quad\quad \wedge \text{IS-LINK}(hg_o,e_o',e')))\\
&\quad \text{IFB}_A(\text{LINK-ERROR}_{BFA}(g_o,e_o,n_o,e_o'),\\
&\qquad \text{IS-LINK}_{BFA}(g_o,e,e'),\\
&\qquad \text{IS-LINK}_{BFA}(g_o,e,e')\\
&\qquad\quad \vee_A(\text{IS-LINK}_{BFA}(g_o,e,e_o)\\
&\qquad\quad\quad \wedge \text{IS-LINK}_{BFA}(g_o,e_o',e')))\\
&= \text{IS-LINK}_{BFA}(g,e,e').
\end{aligned}$$

As in the case "$g = EMPTY_{BFA}$" one gets

$$LINK\text{-}ERROR(hg,e,n,e') \equiv LINK\text{-}ERROR_{BFA}(g,e,n,e').$$

Thus we have shown (*) for $\sigma \in \{$IS-PART, IS-LINK, LINK-ERROR$\}$, and it remains to prove

a) $ADD\text{-}PART(hg,e) \equiv h(ADD\text{-}PART_{BFA}(g,e))$,

b) $ADD\text{-}LINK(hg,e,n,e') \equiv h(ADD\text{-}LINK_{BFA}(g,e,n,e'))$

for all $g \in (BFA)_{\underline{graph}} - \{EMPTY_{BFA}\}$, $e,e' \in A_{\underline{entry}}$ and $n \in A_{\underline{nat}}$.

As to a):

Case 1: $f(ADD\text{-}PART_{BFA}(g,e))$ is a node, say e_o.

Case 1.1: $e_o \neq e$. Then e_o is a node of g, and there is g_o with $ADD\text{-}PART_{BFA}(g_o,e_o) = g$ and $\#(g) = \#(g_o)+1$. By induction hypothesis and (ii),

$$\begin{aligned} ADD\text{-}PART(hg,e) &\equiv ADD\text{-}PART(ADD\text{-}PART(hg_o,e_o),e) \\ &\equiv ADD\text{-}PART(ADD\text{-}PART(hg_o,e),e_o) \\ &\equiv ADD\text{-}PART(h(ADD\text{-}PART_{BFA}(g_o,e)),e_o) \\ &= h(ADD\text{-}PART_{BFA}(ADD\text{-}PART_{BFA}(g_o,e),e_o)) \\ &= h(ADD\text{-}PART_{BFA}(g,e)). \end{aligned}$$

Case 1.2: $e_o = e$. Then a) follows from (ii).

Case 2: $f(ADD\text{-}PART_{BFA}(g,e) = \langle e_o,n_o,e_o' \rangle$.
Then $ADD\text{-}LINK_{BFA}(g_o,e_o,n_o,e_o') = g$ for some g_o with $\#(g_o)+1 = \#(g)$. Hence $LINK\text{-}ERROR_{BFA}(g_o,e_o,n_o,e_o') = FALSE_A$.

<u>Case 2.1</u>. $e \in \{e_o, e_o'\}$. Then e is a node of g_o, and we get by induction hypothesis,

$$\begin{aligned} & \text{ADD-PART}(hg,e) \\ & \equiv \text{ADD-PART}(\text{ADD-LINK}(hg_o,e_o,n_o,e_o'),e) \\ & \equiv \text{ADD-LINK}(hg_o,e_o,n_o,e_o') \equiv hg \\ & = h(\text{ADD-PART}_{BFA}(g,e)). \end{aligned}$$

<u>CASE 2.2</u>: $e \notin \{e_o, e_o'\}$. Then by induction hypothesis and (iii),

$$\begin{aligned} & \text{ADD-PART}(hg,e) \\ & \equiv \text{ADD-PART}(\text{ADD-LINK}(hg_o,e_o,n_o,e_o'),e) \\ & \equiv \text{ADD-LINK}(\text{ADD-PART}(hg_o,e),e_o,n_o,e_o') \\ & \equiv \text{ADD-LINK}(h(\text{ADD-PART}_{BFA}(g_o,e)),e_o,n_o,e_o') \\ & = h(\text{ADD-PART}_{BFA}(g,e)). \end{aligned}$$

<u>As to b)</u>:

<u>Case 1</u>: $f(\text{ADD-LINK}_{BFA}(g,e,n,e'))$ is a node, say e_o. Then e_o is a node of g, and there is g_o with

$$\text{ADD-PART}_{BFA}(g_o,e_o) = g \text{ and } \#(g) = \#(g_o)+1.$$

<u>Case 1.1</u>: $\text{LINK-ERROR}_{BFA}(g,e,n,e') = \text{TRUE}_A$. Then by induction hypothesis,

$$\begin{aligned} & \text{ADD-LINK}(hg,e,n,e') \\ & \equiv \text{ADD-LINK}(\text{ADD-PART}(hg_o,e_o),e,n,e') \\ & \equiv \text{ADD-PART}(hg_o,e_o) \equiv hg \\ & = h(\text{ADD-LINK}_{BFA}(g,e,n,e')). \end{aligned}$$

<u>Case 1.2</u>: $\text{LINK-ERROR}_{BFA}(g,e,n,e') = \text{FALSE}_A$. Since e_o is an isolated node of g, we have $e_o \notin \{e,e'\}$. Thus by induction hypothesis and (ii),

$$\begin{aligned} & \text{ADD-LINK}(hg,e,n,e') \\ & \equiv \text{ADD-LINK}(\text{ADD-PART}(hg_o,e_o),e,n,e') \\ & \equiv \text{ADD-PART}(\text{ADD-LINK}(hg_o,e,n,e'),e_o) \\ & \equiv \text{ADD-PART}(h(\text{ADD-LINK}_{BFA}(g_o,e,n,e')),e_o) \end{aligned}$$

$= h(\text{ADD-LINK}_{BFA}(g,e,n,e'))$.

Case 2: $f(\text{ADD-LINK}_{BFA}(g,e,n,e')) = \langle e_o, n_o, e_o' \rangle$.

Case 2.1: $\langle e,n,e' \rangle = \langle e_o, n_o, e_o' \rangle$. Then b) follows from (iii).

Case 2.2: $\langle e,n,e' \rangle \neq \langle e_o, n_o, e_o' \rangle$. Then $\text{ADD-LINK}_{BFA}(g_o, e_o, n_o, e_o') = g$ for some g_o with $\#(g) = \#(g_o)+1$.

Case 2.2.1: $\text{LINK-ERROR}_{BFA}(g,e,n,e') = \text{TRUE}_A$.
Then by induction hypothesis,

$$\begin{aligned} &\text{ADD-LINK}(hg,e,n,e') \\ &\quad \equiv \text{ADD-LINK}(\text{ADD-LINK}(hg_o,e_o,n_o,e_o'),e,n,e') \\ &\quad \equiv \text{ADD-LINK}(hg_o,e_o,n_o,e_o') \equiv hg \\ &\quad = h(\text{ADD-LINK}_{BFA}(g,e,n,e')). \end{aligned}$$

Case 2.2.2: $\text{LINK-ERROR}_{BFA}(g,e,n,e') = \text{FALSE}_A$.
Then by induction hypothesis and (iii),

$$\begin{aligned} &\text{ADD-LINK}(hg,e,n,e') \\ &\quad \equiv \text{ADD-LINK}(\text{ADD-LINK}(hg_o,e_o,n_o,e_o'), e,n,e') \\ &\quad \equiv \text{ADD-LINK}(\text{ADD-LINK}(hg_o,e,n,e'),e_o,n_o,e_o') \\ &\quad \equiv \text{ADD-LINK}(h(\text{ADD-LINK}_{BFA}(g_o,e,n,e')),e_o,n_o,e_o') \\ &\quad = h(\text{ADD-LINK}_{BFA}(g,e,n,e')). \end{aligned}$$

Thus we have shown conditions a) and b), i.e. the remaining cases of (*) which had to be proved in order to conclude from Thm. 1.15 that $\langle BPAR,K \rangle$ is correct w.r.t. BF.
By definition of BF, $U_{BPAR} \circ BF(A) = A$ for all $A \in K$. Correctness of $\langle BPAR,K \rangle$ w.r.t. BF implies $BF(A) \cong F_{BPAR}(A)$ for all $A \in K$. Hence BPAR is persistent

w.r.t. K. Thus we infer from Prop. 2.16 that BPAR is a correct extension of <<entry,entry>,K>. In Ex. 8.16 we have shown that this property is sufficient for <u>BSPEC(A)</u> to be <u>a correct extension of</u> <u>bool</u> where $A \in K$.

8.14 <u>Example</u> (parts-system1)

Let BPAR and K be as in Ex. 8.13. Moreover, PAR = <entry,parts-system1> where

```
parts-system1 = parts-system +
     opns: ATOMS: graph entry nat ⟶ bag
     eqns: ATOMS(EMPTY,x,n) = ∅                              p1
           ATOMS(ADD-PART(g,x),y,n)
            = IFC(EQE(x,y)∧¬IS-PART(g,x),
                  INS(∅,y,n),
                  ATOMS(g,y,n))                               p2

           ATOMS(ADD-LINK(g,x,n,y),z,m)
            = IFC(LINK-ERROR(g,x,n,y)
                    ∨¬IS-LINK(g,z,x),
                  ATOMS(g,z,m),
                  IFC(EQE(z,x),
                      JOIN(ATOMS(g,y,n·m),
                           DEL(x,ATOMS(g,x,m)),
                      ATOMS(ADD-LINK(g,x,n,y),z,m)))
                                                              p3
```

In order to show that $R = \{$<u>p1</u>, <u>p2</u>, <u>p3</u>$\}$ is conditionally decreasing, we define for all $A \in K$ a weight function

$$w_A = w_{ATOMS,A}: BG(A)_{\underline{graph}\ \underline{entry}\ \underline{nat}} \longrightarrow \mathbb{N}$$

by

$$w_A(t,u,v) = \begin{cases} 2\cdot n(t)+1 & \text{if } t = \text{ADD-LINK}(t',u',v,u") \\ & \text{and } \text{EQE}(u,u')_A = \text{FALSE}_A \\ 2\cdot n(t) & \text{otherwise} \end{cases}$$

where <u>n(t)</u> is the number of EMPTY-, ADD-PART- and ADD-LINK-symbols in t.
w_A is defined in a way such that the conditions preceding the "recursive calls" of ATOMS on the righthand side of <u>p2</u>- resp. <u>p3</u>-instances imply that the recursive call arguments are BSPEC(A)-congruent to some term tupel which is "w_A-smaller" than the lefthand side arguments. Concerning <u>p3</u> this is expressed by the following lemma.

8.14A <u>Lemma</u>

Let $A \in K$, $t \in BG(A)_{\underline{graph}}$, $u,u',u" \in BG(A)_{\underline{entry}}$ and $v \in BG(A)_{\underline{nat}}$.
Then

(i) LINK-ERROR(t,u,v,u') $\equiv$ FALSE,

(ii) IS-LINK(t,u",u) $\equiv$ TRUE

and

(iii) EQE(u",u) $\equiv$ FALSE

imply $t_o :=$ ADD-LINK(t,u,v,u') $\equiv t_o'$ for some t_o' with $w_A(t_o',u",v) < w_A(t_o,u",v)$.
($\equiv$ denotes $\equiv_{BSPEC(A)}$.)

<u>Proof:</u>

By definition of BFA, we have $n(t) \geq \#(t_{BFA})$ for all $t \in BG(A)_{\underline{graph}}$ (cf. 8.14), while the definition of h in 8.13 implies that for all $g \in (BFA)_{\underline{graph}}$, $n(hg) = \#(g)$. From (i) - (iii) we conclude that there are

$$g \in (BFA)_{\underline{graph}}, \; n \in A_{\underline{nat}} \text{ and } e \in A_{\underline{entry}}$$
$$\text{with } \#(t_{BFA}) = \#(g)+1,$$
$$t_{BFA} = \text{ADD-LINK}_{BFA}(g, u''_{BFA}, n, e)$$

and

$$t_{o,BFA} = \text{ADD-LINK}_{BFA}(\text{ADD-LINK}_{BFA}(g, u_{BFA}, v_{BFA}, u'_{BFA}), u''_{BFA}, n, e).$$

Hence by 8.13(*),

$$t \equiv ht_{o,BFA}$$
$$\equiv \text{ADD-LINK}(\text{ADD-LINK}(hg, u, v, u'), u'', n, e) =: t_o$$

and thus

$$w_A(t_o', u'', v) = 2n(t_o') = 2n(hg)+4 = 2\#(g)+4$$
$$< 2\#(t_{BFA})+3 \leq 2n(t)+3 = 2n(t_o)+1 =$$
$$= w_A(t_o, u'', v). \; \square$$

8.15 Example (parts-system1)

We continue Ex. 8.14 and show that $R = \{\underline{p1}, \underline{p2}\}$ is conditionally decreasing using Prop. 8.12.

Let $A \in K$ and $f \in BZ(G(A))$. For $\langle l,r \rangle = \underline{p1}$ we obtain $pred_{l,f}(r, \text{TRUE}) = true$ because $r = \text{EMPTY} \in \text{BOP-C}$. Let $\langle l,r \rangle = \underline{p2}$ and $p = \text{EQE}(x,y) \wedge \neg\text{IS-PART}(g,x)$. Then

$$pred_{l,f}(r, \text{TRUE})$$
$$= pred_{l,f}(\text{INS}(\emptyset, y, n), \text{TRUE} \wedge p)$$
$$\cdot pred_{l,f}(\text{ATOMS}(g,y,n), \text{TRUE} \wedge \neg p)$$
$$= pred_{l,f}(\emptyset, \text{TRUE} \wedge p) \cdot pred_{l,f}(y, \text{TRUE} \wedge p)$$
$$\cdot pred_{l,f}(n, \text{TRUE} \wedge p)$$

because ATOMS(g,y,n) is simple and

$w_A(fg,fy,fn) < w_A(f \circ arg(l))$

= true.

Let $\langle l,r \rangle$ = <u>p3</u> and p = LINK-ERROR(g,x,n,y) $\vee\neg$IS-LINK(g,z,x).

Then

$pred_{l,f}(r, TRUE)$

$= pred_{l,f}(ATOMS(g,z,m), TRUE \wedge p)$

$\cdot\, pred_{l,f}(IFC(EQE(z,x),\ JOIN(ATOMS(g,y,n\ m),\ DEL(x, ATOMS(g,x,m))),\ l), TRUE \wedge \neg p)$

$= pred_{l,f}(JOIN(ATOMS(g,y,n\ m),\ DEL(x, ATOMS(g,x,m))),\ (TRUE \wedge \neg p) \wedge EQE(z,x))$

$\cdot\, pred_{l,f}(l, (TRUE \wedge \neg p) \wedge \neg EQE(z,x))$

because ATOMS(g,z,m) is simple and

$w_A(fg,fz,fm) < w_A(f \circ arg(l))$

$= pred_{l,f}(ATOMS(g,y,n\ m), (TRUE \wedge \neg p) \wedge EQE(z,x))$

$\cdot pred_{l,f}(DEL(x, ATOMS(g,x,m)), (TRUE \wedge \neg p) \wedge EQE(z,x))$

because l is simple and since

$f((TRUE \wedge \neg p) \wedge EQE(z,x)) \equiv TRUE$ implies

$w_A(t) < w_A(f \circ arg(l))$ for some $t \epsilon BG(A)$[3]

with $t \equiv f \circ arg(l)$

$= pred_{l,f}(x, (TRUE \wedge \neg p) \wedge EQE(z,x))$

$\cdot pred_{l,f}(ATOMS(g,x,m), (TRUE \wedge \neg p) \wedge EQE(z,x))$

because $ATOMS(g,y,n \cdot m)$ is simple and

$w_A(fg,fy,f(n \cdot m)) < w_A(f \circ arg(l))$

= true

because ATOMS(g,x,m) is simple and

$w_A(fg,fx,fm) < w_A(f \circ arg(l))$.

Thus by Prop. 8.12, R = {p1, p2, p3} is conditionally decreasing. □

8.16 Completeness Theorem

Let SCCR be the set of conditional-compatibility rules with simple lefthand side (cf. 8.3). If E contains a subrelation R such that R is base-total w.r.t. BOP-C and R∪SCCR is conditionally decreasing, then PAR is complete w.r.t. <BPAR,K>.

Proof:

First we show that PAR is complete w.r.t. <BPAR,K> if

(*) for all $A \in K$ and all simple terms $t \in G(A)$ there is $t' \in BG(A)$ with $t \equiv_{CSPEC(A)} t'$

where CSPEC = <S,OP,E∪SCCR>: Let $t \in G(A)$. The existence of $t' \in BG(A)$ with $t \equiv_{SPEC(A)} t'$ is proved by induction on the number n(t) of (OP-BOP)-symbols occurring in t:
IF n(t) = 0, we are done with t' = t. Otherwise there is a simple subterm u of t, which by (*), is CSPEC(A)-congruent to some $u' \in BG(A)$. Replacing u in t by u', we obtain t_1 with $t \equiv_{CSPEC(A)} t_1$ and $n(t_1) < n(t)$. By induction hypothesis, some $t' \in BG(A)$ satisfies $t_1 \equiv_{SPEC(A)} t'$, and thus $t \equiv_{CSPEC(A)} t'$ .Now there are a least number k and $t_1, \ldots, t_k \in G(A)$ such that $t_1 = t$, $t_n = t'$ and for all $1 \leq i < k$,

$$t_i \xleftrightarrow[E(A) \cup SCCR]{} t_{i+1}.$$

By induction on k, we get $t \equiv_{SPEC(A)} t'$: k = 0 contradicts n(t)>0 = n(t'). If k>0, then $t_2 \equiv_{SPEC(A)} t'$ by induction hypothesis, and either $t \xleftrightarrow[E(A)]{} t_2$ or $t \xleftrightarrow[SCCR]{} t_2$. In the first case we are done. In the second case there are $<l,r> \in SCCR$ and $f \in Z(G(A))$ such that w.l.o.g. $fl = t$ and $fr = t_2$. Hence $n(fx) < n(t)$ for all $x \in var(l) = var(r)$. Thus our first induction hypothesis implies

$$t \equiv_{SPEC(A)} gl \text{ and } t_2 \equiv_{SPEC(A)} gr$$

for some $g \in BZ(G(A))$. 8.5(a) yields $gl \equiv_{SPEC(A)} gr$ so that $t \equiv_{SPEC(A)} t_2$, and we conclude $t \equiv_{SPEC(A)} t'$ in the second case, too.

Therefore, completeness of PAR follows from (*) and thus it remains to show (*).

Let $A \in K$. For all $f \in BZ(G(A))$, we define a predicate

$$q_f : ST \times BP \longrightarrow \{true, false\}$$

(cf. 8.10) by

$$\begin{aligned} q_f(t,p) &= true \\ \text{iff } fp &\equiv_{BSPEC(A)} TRUE \text{ implies} \\ ft &\equiv_{CSPEC(A)} t' \text{ for some } t' \in BG(A). \end{aligned}$$

Suppose that for all simple terms $l \in T$, $f \in BZ(G(A))$, $t \in ST$ and $p \in BP$

(**) $pred_{l,f}(t,p) = true$ implies $q_f(t,p) = true$.

The step from (**) to (*) proceeds as follows:

Let $l \in G(A)$ be simple. Since R is base-total w.r.t. BOP-C, we conclude from the proof of Thm. 8.5(b) that $R \cup CCR$ is base-total. Hence $l = fl'$ for some $<l',r'> \in R \cup SCCR$ and $f \in BZ(G(A))$. Since $R \cup SCCR$ is conditionally decreasing, $pred_{l',f}(r',TRUE) = true$, and thus (**) implies

$$l = fl' \equiv_{CSPEC(A)} fr' \equiv_{CSPEC(A)} t'$$

for some $t' \in BG(A)$.

Finally, we have to prove (**).

Let $l \in T$ be simple, $f \in BZ(G(A))$, $t \in ST$ and $p \in BP$ such that $pred_{l,f}(t,p)$ = true. We show $q_f(t,p)$ = true by induction on $\langle w_{\sigma,A}(f \circ arg(l)), size(t)\rangle$ with respect to the lexicographic order $>$ on $\mathbb{N}^2$. From the definition of $pred_{l,f}$ (8.12) we deduce four cases:

Case 1: t is simple and dec(t,p) holds (cf. 8.10).

Case 1.1: $w_{root(l),A}(f \circ arg(l)) = 0$. Then $fp \equiv_{BSPEC(A)} TRUE$ would contradict dec(t,p). Hence $q_f(t,p)$ = true.

Case 1.2: $w_{root(l),A}(f \circ arg(l)) > 0$. Let $fp \equiv_{BSPEC(A)} TRUE$. Since dec(t,p) holds, there is $t' \in BG(A)^*$ such that $t' \equiv_{BSPEC(A)} f \circ arg(t)$ and

$$w_{root(t),A}(t') < w_{root(l),A}(f\ arg(l))$$

Let $\sigma = root(t)$. Since R is base-total w.r.t. BOP-C, $R \cup CCR$ is base-total (see the proof of Thm. 8.5(b)). Thus $\sigma t' = gl'$ for some $\langle l', r'\rangle \in R \cup SCCR$ and $g \in BZ(G(A))$. Since $R \cup SCCR$ is conditionally decreaseing, $pred_{l',g}(r', TRUE)$ = true. By induction hypothesis, $q_g(r', TRUE)$ = true, i.e. $gr' \equiv_{CSPEC(A)} t''$ for some $t'' \in BG(A)$.

Hence

$$ft = \sigma(f \circ arg(t)) \equiv_{SPEC(A)} \sigma t' = gl' \equiv_{CSPEC(A)} gr' \equiv_{CSPEC(A)} t''.$$

Therefore, $q_f(t,p)$ = true.

Case 2: t is simple and dec(t,p) does not hold. Then $pred_{l,f}(t,p)$ = false, which contradicts our assumption.

Case 3: $t = IF(q,u,u')$ for some $IF \in C$ and $q \in BP$. Then $pred_{1,f}(u,p \wedge q) = pred_{1,f}(u',p \wedge \neg q) = true$ so that $q_{1,f}(u,p \wedge q) = q_{1,f}(u',p \wedge \neg q) = true$ by induction hypothesis. Let $fp \equiv_{BSPEC(A)} TRUE$ and w.l.o.g. $fq \equiv_{BSPEC(A)} TRUE$. Hence $q_{1,f}(u,p \wedge q) = true$ implies $fu \equiv_{SPEC(A)} t'$ for some $t' \in BG(A)$. Therefore,

$$ft \equiv_{SPEC(A)} IF(TRUE,fu,fu') \equiv_{SPEC(A)} fu \equiv_{SPEC(A)} t',$$

and thus $q_f(t,p) = true$.

Case 4: Neither of cases 1 - 3 hold.

Case 4.1: $t \in BOP \cup X$. Then $ft \in BG(A)$ so that $q_f(t,p) = true$.

Case 4.2: $root(t) \in BOP$ and for all $1 \leq i \leq lg(arg(t))$, $pred_{1,f}(arg(t)_i,p) = true$. Thus $q_{1,f}(arg(t)_i,p) = true$ by induction hypothesis. Hence $fp \equiv_{BSPEC(A)} TRUE$ implies $f(arg(t)_i) \equiv_{SPEC(A)} t_i$ for some $t_i \in BG(A)$. Let $\sigma = root(t)$. Then

$$ft = \sigma(f \circ arg(t)) \equiv_{SPEC(A)} \sigma(t_1,\ldots,t_n) \in BG(A).$$

Therefore, $q_f(t,p) = true$. □

8.17 Example (parts-system1)

Again we refer to Ex. 8.14, now in order to show that PAR is complete w.r.t. <BPAR,K> using Thm. 8.16. Thus we have to prove that $R = \{p1, p2, p3\}$ is base-total w.r.t. BOP-C and that $R \cup SCCR$ is conditionally decreasing.

R is base-total w.r.t. BOP-C if M:=arg(R,ATOMS) is <u>graph</u> <u>entry</u> <u>nat</u>-generating w.r.t. BOP-C (cf. Def. 8.4). Since

$$M = \{\langle EMPTY,x,n\rangle, \langle ADD\text{-}PART(g,x),y,n\rangle, \langle ADD\text{-}LINK(g,x,n,y),z,m\rangle\},$$

the computation of p(M) (cf. 7.3) results in

$$p(M) = p(\{\langle x,n\rangle\})\cdot p(\{\langle g,x,y,n\rangle\})\cdot p(\{\langle g,x,n,y,z,m\rangle\}) = true.$$

Hence by Thm. 7.3, M is <u>graph</u> <u>entry</u> <u>nat</u>-generating w.r.t. BOP-C.

It was shown in Ex. 8.15 that R = {<u>p1</u>, <u>p2</u>, <u>p3</u>} is conditionally decreasing with respect to the weight function $w_{ATOMS,A} = w_A$ given in Ex. 8.14. Hence by Prop. 8.12, R∪SCCR is conditionally decreasing if for all <l,r>∈SCCR and f∈BZ(G(A)) $pred_{l,f}(r,TRUE)$ = true.

So let <l,r>∈SCCR. Then

$$l = ATOMS(IFG(b,x,y),e,n)$$

and

$$r = IFC(b,ATOMS(x,e,n),ATOMS(y,e,n))$$

for some b,x,y,e,n∈X. Thus

$$pred_{l,f}(r,TRUE)$$
$$= pred_{l,f}(ATOMS(x,e,n),TRUE\wedge b)$$
$$\cdot pred_{l,f}(ATOMS(y,e,n),TRUE\wedge\neg b) = true$$

because ATOMS(x,e,n) and ATOMS(y,e,n) are simple and $w_A(fx,fe,fn), w_A(fy,fe,fn) < w_A(f\circ arg(l))$. Therefore we conclude from Thm. 8.16 that PAR is complete w.r.t. <BPAR,K>. □

Two classes of conditionally decreasing relations are defined in the following. The first one supposes a mapping $BF : K \longrightarrow Alg(BSPEC)$ such that $\langle BPAR,K\rangle$ is correct w.r.t. BF and uses a right-inverse of the term evaluation (cf. Thm. 1.15). The second class consists of recursive equations for search operations and corresponds to function definitions by bounded minimization (Brainerd, Landweber /8/, p. 64).

8.18 Definition

Let $\langle BPAR,K\rangle$ be correct w.r.t. some $BF : K \longrightarrow Alg(BSPEC)$. A conditionally decreasing relation R is semantically decreasing if for all $\sigma \in OP-BOP$ and $A \in K$ the weight function

$$w_{\sigma,A}: BG(A)_{arity(\sigma)} \longrightarrow \mathbb{N}$$

is defined by

$$w_{\sigma,A}(t) = size(gt_{BFA})$$

where g is a right-inverse of the term evaluation $eval_{BFA}: BG(A) \longrightarrow BFA$ that satisfies

$$\sigma g(a) \equiv_{BSPEC(A)} g\sigma_{BFA}(a)$$

for all $w \in BS^*$, $s \in BS$, $\sigma \in BOP(A)_{w,s}$ and $a \in (BFA)_w$. (cf. Thm. 1.15(ii)).

8.18A Example (nat1)

Let

$$nat1 = nat +$$

$$opns: FACT: \underline{nat} \rightarrow \underline{nat}$$

eqns: FACT(x) = IFN(EQN(x,0),S0,x·FACT(Px)) <u>f1</u>

(cf. 1.3), PSPEC = <∅,∅,∅>, BPAR = <PSPEC,nat>, PAR = <PSPEC, nat1> and K = {∅}. We choose BF∅ ∈ Alg(BSPEC) with $BF\emptyset_{\underline{bool}}$ = {true,false}, $BF\emptyset_{\underline{nat}} = \mathbb{N}$ and corresponding operations. A right-inverse g of the term evaluation $eval_{BF0}: BG \to BF\emptyset$ is given by g(TRUE) = true, g(FALSE) = false, g0 = 0 and g(n+1) = S(gn). A simple induction yields

$$\sigma g(a) \equiv_{nat} g\sigma_{BF\emptyset}(a)$$

for all w∈BS*, s∈BS, $\sigma \in BOP_{w,s}$ and $a \in BF\emptyset_w$. Hence by Thm. 1.15, <BPAR,K> is correct w.r.t. BF.

Let C = {IFN}. With $w = w_{FACT,\emptyset}: BG_{\underline{nat}} \to \mathbb{N}$ defined by $w(t) = size(gt_{BF\emptyset})$, R = {<u>f1</u>} becomes conditionally decreasing:

By Prop. 8.12, we have to show that for all <l,r>∈R and f∈BZ(G) $pred_{l,f}(r,TRUE) = true$.
So let <l,r> = <u>f1</u> and p = EQN(x,0). Then

$$\begin{aligned}
& pred_{l,f}(r,TRUE) = \\
&= pred_{l,f}(S0,TRUE\wedge p)\cdot pred_{l,f}(x\cdot FACT(Px),TRUE\wedge\neg p) \\
&= pred_{l,f}(0,TRUE\wedge p)\cdot pred_{l,f}(x,TRUE\wedge\neg p) \\
&\quad \cdot pred_{l,f}(FACT(Px),TRUE\wedge\neg p) \\
&= true
\end{aligned}$$

because FACT(Px) is simple and $f(TRUE\wedge\neg p) \equiv_{BSPEC} TRUE$ implies $(fx)_{BF\emptyset} \neq 0$ and thus

$$\begin{aligned}
& w(fPx) = size(g(fPx)_{BF\emptyset}) < size(Sg(fPx)_{BF\emptyset}) \\
& = size(g((fPx)_{BF\emptyset}+1)) = size(g((Pfx)_{BF\emptyset}+1)) \\
& = size(g(fx)_{BF\emptyset}) = w(fx) = w(f\circ arg(l)).
\end{aligned}$$

Therefore by Def. 8.17, {fl} is semantically decreasing.

8.19 Example (expression)

```
Let
   ranked-entry = entry +
        opns: RANK: entry → nat,

   tree = ranked-entry +
        sorts: tree, treelist
        opns:  MAKE: entry treelist → tree
               ENTRY: tree → entry
               SUCC: tree → treelist
               NIL: → treelist
               APPEND: tree treelist → treelist
               FIRST: treelist → tree
               REST: treelist → treelist
               LENGTH: treelist → nat
        eqns:  ENTRY(MAKE(x,l)) = x
               SUCC(MAKE (x,l)) = l
               FIRST(NIL) = MAKE(UNDEF,NIL)
               FIRST(APPEND(t,l)) = t
               REST(NIL) = NIL
               REST(APPEND(t,l)) = l
               LENGTH(NIL) = 0
               LENGTH(APPEND(t,l)) = S(LENGTH(l))
and
   expression = tree +
        opns:  IS-EXPR: tree → bool
               IS-EXPRLIST: treelist → bool
        eqns:  IS-EXPR(x)
                  = EQN(RANK(ENTRY(x)),LENGTH(SUCC(x)))
                    ∧IS-EXPRLIST(SUCC(x))                        ex1
               IS-EXPRLIST(x)
                  = IFB(EQN(LENGTH(x),0),
                        TRUE,
                        IS-EXPR(FIRST(x))
                         IS-EXPRLIST(REST(x)))                   ex2
```

Let BPAR = <ranked-entry,tree>,
PAR = <ranked-entry,expression> and K be the class of all ranked-entry algebras A which satisfy 1.11 (i) and (ii). We choose $BF: K \longrightarrow Alg(BSPEC)$ such that $(BFA)_s = A_s$ for all $s \in PS$;

$$(BFA)_{\underline{tree}} = \text{set of all finite trees with nodes in } A_{\underline{entry}};$$

$$(BFA)_{\underline{treelist}} = (BFA)^{*}_{\underline{tree}}; \quad \sigma_{BFA} = \sigma_A \text{ for all } \sigma \in POP,$$

MAKE, NIL, APPEND are interpreted as tree resp. list constructors; ENTRY, FIRST, SUCC, REST provide projections; and $LENGTH_{BFA}$ computes the length of lists. A right-inverse g of the term evaluation $eval_{BFA}: BG(A) \longrightarrow BFA$ is given by

$$g_s = id \quad \text{for all } s \in PS,$$
$$g(MAKE_{BFA}(e,l)) = MAKE(e,gl),$$
$$g(NIL_{BFA}) = NIL$$

and

$$g(APPEND_{BFA}(b,l)) = APPEND(gb,gl).$$

A simple induction yields

$$\sigma g(a) \equiv_{tree(A)} g\sigma_{BFA}(a)$$

for all $w \in BS^*$, $s \in BS$, $A \in K$, $\sigma \in BOP(A)_{w,s}$ and $a \in (BFA)_w$.
Hence by Thm. 1.15, <BPAR,K> is correct w.r.t. BF.

Let $C = \{IFB\}$, $A \in K$ and $\sigma \in \{$IS-EXPR, IS-EXPRLIST$\}$.
With $w_{\sigma,A}$ defined as in Def. 8.17, $R = \{\underline{ex1}, \underline{ex2}\}$

becomes conditionally decreasing:

By Prop. 8.12, we have to show that for all $\langle l,r\rangle \in R$, $A \in K$ and $f \in BZ(G(A))$, $pred_{l,f}(r,TRUE) = $ $= true$.

So let $\langle l,r\rangle = \underline{ex1}$. Then

$$\begin{aligned}
& pred_{l,f}(r,TRUE) \\
&= pred_{l,f}(EQN(RANK(ENTRY(x),LENGTH(SUCC(x)),TRUE) \\
&\quad \cdot pred_{l,f}(IS\text{-}EXPRLIST(SUCC(x)),TRUE) \\
&= true
\end{aligned}$$

because IS-EXPRLIST(SUCC(x)) is simple and

$$\begin{aligned}
& w_{IS\text{-}EXPRLIST,A}(fSUCCx) \\
&= size(g(fSUCCx)_{BFA}) \\
&< size(MAKE\ (e,g(fSUCCx)_{BFA})) \\
&= size(gMAKE_{BFA}(e,(fSUCCx)_{BFA})) \\
&= size(gMAKE_{BFA}(e,(SUCCfx)_{BFA})) \\
&= size(g(fx)_{BFA}) = w_{IS\text{-}EXPR,A}(fx) \\
&= w_{IS\text{-}EXPR,A}(f\circ arg(l))
\end{aligned}$$

for some $e \in A_{\underline{entry}}$.

Let $\langle l,r\rangle = \underline{ex2}$ and $p = EQN(LENGTH(x),0)$. Then

$$\begin{aligned}
& pred_{l,f}(r,TRUE) \\
&= pred_{l,f}(TRUE,TRUE\wedge p) \\
&\quad \cdot pred_{l,f}(IS\text{-}EXPR(FIRST(x))\wedge IS\text{-}EXPRLIST(REST(x)),TRUE\wedge\neg p) \\
&= pred_{l,f}(IS\text{-}EXPR(FIRST(x)),TRUE\wedge\neg p) \\
&\quad \cdot pred_{l,f}(IS\text{-}EXPRLIST(REST(x)),TRUE\wedge\neg p) \\
&= true
\end{aligned}$$

because IS-EXPR(FIRST(x)) and IS-EXPRLIST(REST(x)) are simple and $f(TRUE\wedge\neg p) \equiv_{BSPEC(A)} TRUE$ implies $(fx)_{BFA} \neq \varepsilon$ and thus

$$\left.\begin{array}{l} w_{IS\text{-}EXPR,A}(fFIRSTx) = size(g(fFIRSTx)_{BFA}) \\ w_{IS\text{-}EXPRLIST,A}(fRESTx) = size(g(fRESTx)_{BFA}) \end{array}\right\}$$

$$\begin{aligned} &< size(APPEND(g(fFIRSTx)_{BFA}, g(fRESTx)_{BFA})) \\ &= size(gAPPEND_{BFA}((fFIRSTx)_{BFA}, (fRESTx)_{BFA})) \\ &= size(gAPPEND_{BFA}((FIRSTfx)_{BFA}, (RESTfx)_{BFA})) \\ &= size(g(fx)_{BFA}) = w_{IS\text{-}EXPRLIST,A}(fx) \\ &= w_{IS\text{-}EXPRLIST,A}(f \circ arg(1)). \end{aligned}$$

Therefore by Def. 8.17, {<u>ex1</u>, <u>ex2</u>} is semantically decreasing. □

The second class of conditionally decreasing relations mentioned above only admits rules of a special form:

8.20 <u>Proposition</u>

Let <l,r> be a rule such that

$$l = \sigma x \quad \text{and} \quad r = IF(p,t,\sigma hx)$$

for some $\sigma \in OP\text{-}BOP$, $x \in BX^*$, $IF \in C$, $p,t \in BT$ and $h \in BZ(T)$. If for all $A \in K$ and $f \in BZ(G(A))$ there is $k \in \mathbb{N}$ with

$$fh^k p \equiv_{BSPEC(A)} TRUE,$$

then {<l,r>} is conditionally decreasing. <l,r> is called a <u>minimization rule</u>.

<u>Proof:</u>

Let $A \in K$ and $w_{\sigma,A}: BG(A)_{arity(\sigma)} \longrightarrow \mathbb{N}$ be defined by

$$w_{\sigma,A}(t) = \min\{k\in N/fh^k p \equiv_{BSPEC(A)} TRUE\}$$

where fx = t. Hence for all $f\in BZ(G(A))$

$pred_{1,f}(r,TRUE) = pred_{1,f}(t,TRUE\wedge p)\cdot pred_{1,f}(\sigma hx,TRUE\wedge\neg p)$
= true

because σhx is simple and $f(TRUE\wedge\neg p) \equiv_{BSPEC(A)} TRUE$ implies $fp \equiv_{BSPEC(A)} TRUE$ and thus

$$\begin{aligned} w_{\sigma,A}(fhx) &= \min\{k\ N/fh^{k+1}p \equiv_{BSPEC(A)} TRUE\} \\ &< \min\{k\in N/fh^k p\ \ _{BSPEC(A)} TRUE\} = w_{\sigma,A}(fx) \\ &= w_{\sigma,A}(f\circ arg(1)). \end{aligned}$$

Thus by Prop. 8.12, $\{<1,r>\}$ is conditionally decreasing. □

8.21 Example (array2)

Let BPAR = <entry,array1> (cf. Ex. 2.2),

```
array2 = array1 +
    opns: SEARCHSLOT: array nat → nat
          PACK: array nat → array
    eqns: SEARCHSLOT(a,n) = IFN(EQE(GET(a,n),UNDEF),
                                n,
                                SEARCHSLOT(a,Sn))       a7

          PACK(a,n) = IFN(EQE(GET(a,Sn),UNDEF),
                          PUT(a,n,GET(a,Sn)),
                          PACK(PUT(a,n,GET(a,Sn)),Sn))
                                                        a8
```

and K be the class of entry-algebras defined in Ex. 8.13 and equipped with the additional condition that for all $n\in A_{\underline{nat}}$ $S_A n$ differs from n.

In Ex. 2.2, we defined a mapping $F:K \to Alg(array1)$ and proved that $\langle BPAR,K\rangle$ is correct w.r.t. F. By Prop. 8.20, $\{\underline{a7},\underline{a8}\}$ is conditionally decreasing if for all $A\in K$ and $f\in BZ(G(A))$

(i) $EQE(GET(fa,fS^k n),UNDEF) \equiv_{BSPEC(A)} TRUE$ for some $k\in\mathbb{N}$,

(ii) $EQE(GET(fh^k a,fS^{k+1} n),UNDEF) \equiv_{BSPEC(A)} TRUE$ for some $k\in\mathbb{N}$ where $ha = PUT(a,n,GET(a,Sn))$ and $hn = Sn$.

Since for all $g\in(FA)_{\underline{array}}$ (cf. 1.11) there are only finitely many $n\in A_{\underline{nat}}$ with $an \neq UNDEF_A$, we get for all $m\in A_{\underline{nat}}$ some $k\in\mathbb{N}$ with $g(S^k{}_A m) = UNDEF_A$. Hence

$$EQE(GET(fa,S^k fn),UNDEF)_{FA} = TRUE_A$$

where $(fa)_{FA} = g$ and $(fn)_A = m$. Thus correctness of $\langle BPAR,K\rangle$ w.r.t. F implies (i). By the same argument, for all $f\in BZ(G(A))$ some $k\in\mathbb{N}$ satisfies

$$EQE(GET(fa,fS^{k+1} n),UNDEF) \equiv_{BSPEC(A)} TRUE. \qquad (*)$$

A simple induction on k yields for all $i>k$

$$GET(fh^k a,fS^i n) \equiv_{BSPEC(A)} GET(fa,fS^i n) \qquad (**)$$

where $ha = PUT(a,n,GET(a,Sn))$ and $hn = Sn$. Therefore, (ii) follows from (*) and (**).

9. Critical term pairs and absolute confluence

The main assumption of Consistency Theorem 4.19 is the existence of term relations $R(A)$, $A \in K$, such that $\langle R(A), A \rangle$ is absolutely confluent (cf. 4.15). In this chapter we give a "local" characterization of absolute confluence. Replacing $G(A)$ by $T(A)$ in Def. 4.15, we obtain the usual notion of confluence referred to in Huet /30/ and Knuth, Bendix /43/. Since confluence is stronger than absolute confluence, the characterization of confluence given in Huet /30/, Thm. 3.2 contains a stronger local condition than Thm. 9.15 below does. Both theorems require that the term relation in question is normalizing (cf. 4.4). If this is not the case, we have to assume that the term relation is non-ambiguous (lefthand sides do not overlap, cf. 4.13), which is an essentially stronger property than the local condition in Thm. 9.15. This condition uses the notion of a critical term pair (Def. 9.3).

9.1 Definition

Let $A \in K$, $t, t' \in T(A)^*$, $f \in Z(T(A))$ and $\beta: X \longrightarrow X$ be an injective mapping. The pair $\langle f, \beta \rangle$ is called a <u>unifier of $\langle t, t' \rangle$</u> if $f\beta t = ft'$ and $var(\beta t) \cap var(t') = \emptyset$. If for all unifiers $\langle g, \beta \rangle$ of $\langle t, t' \rangle$ there is $h \in Z(T(A))$ with $hf = g$ and if $fx = x$ for all $x \in X \cap f(X)$, then $\langle f, \beta \rangle$ is a <u>most general unifier of $\langle t, t' \rangle$</u>.

If $\langle t, t' \rangle \in (T(A)^*)^2$ is unifiable, then $\langle t, t' \rangle$ has a most general unifier (cf. Robinson /59/, 5.9), which is unique up to renaming of finitely many variables:

9.2 Lemma

Let $<f,\beta>$ and $<g,\beta>$ be two most general unifiers of $<t,t'>$. Then $\alpha f = g$ for some $\alpha: X \longrightarrow X$.

Proof:

By assumption, there are $h,h' \in Z(T(A))$ with $hf = g$ and $h'g = f$. Hence $h'hf = f$, i.e. for all $x \in X$ $h'h(fx) = fx$. Therefore, $h'hx = x$ for all $x \in var(fX)$, and thus $hx \in X$ for all $x \in var(fX)$. We choose $\alpha: X \longrightarrow X$ such that $\alpha x = hx$ for all $x \in var(fX)$ and obtain $\alpha f = g$. $\square$

9.3 Definition

Let $A \in K$ and $R, R' \subseteq T(A)^2$.

R_X denotes the set of all pairs $<\alpha l, \alpha r>$ where $<l,r> \in R$ and $\alpha: X \longrightarrow X$ is an injective mapping.

Let $<l,r> \in R$, $u \in T(A)$ and $g \in Z(T(A))$ such that

(i) $\quad l = gu$.

Suppose that $\{x_1, \ldots, x_n\} := \{x \in var(u) / gx \notin X\}$ is nonempty and there are $<l_1,r_1>, \ldots, <l_n,r_n> \in R'_X$ such that

(ii) $\quad <<gx_1, \ldots, gx_n>, <l_1, \ldots, l_n>>$ has a most general unifier $<f,\beta>$.

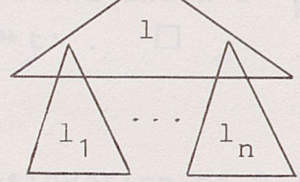
Let $h \in Z(T(A))$ be defined by

$$hx = \begin{cases} r_i & \text{if } x = x_i \text{ for some } 1 \leq i \leq n \\ \beta g x & \text{otherwise.} \end{cases}$$

Then the pair $\langle t,t'\rangle := \langle f\beta r, fhu\rangle$ is critical w.r.t. $\langle R,R'\rangle$ and $\langle\langle l,r\rangle, f\beta\rangle$ is a generator of $\langle t,t'\rangle$.

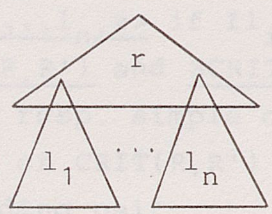
9.3 (i) and (ii) describe formally that $l_1,\ldots,l_n$ overlap l.
The "overlapped term" is given by $f\beta l$:

$f\beta l$ =

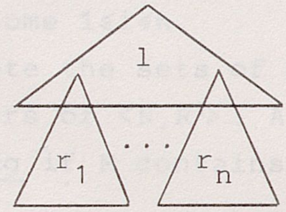


t and t' are the terms which result from $f\beta l$ by applying $\langle l,r\rangle$ resp. $\langle l_1,r_1\rangle,\ldots,\langle l_n,r_n\rangle$ to $f\beta l$, i.e.

t =



and t' =



This is confirmed by the following proposition.

9.4 Proposition

Let $\xrightarrow[R']{}\!\!\!\!\rightarrow$ be the least OP(A)-stable and parallel OP(A)-compatible relation on T(A) that includes R' (cf. 10.5). Then

$$\begin{tikzcd} & t \\ f\beta l \arrow[ur, "R"] \arrow[dr, two heads, "R'"'] & \\ & t' \end{tikzcd}$$

Proof:

Clearly, $f\beta l \xrightarrow[R]{} f\beta r$. Define $h' \in Z(T(A))$ by

$$h'x = \begin{cases} l_i & \text{if } x = x_i \text{ for some } 1 \leq i \leq n \\ \beta g x & \text{otherwise.} \end{cases}$$

Then $fh'x_i = fl_i = f\beta gx_i$ for all $1 \leq i \leq n$ and thus $fh'x = f\beta gx$ for all $x \in X$. Hence $f\beta l = f\beta gu = fh'u$. Since $t' = fhu$ and $h'x \xrightarrow[R']{\Delta} hx$ for all $x \in X$, we get $f\beta l \xrightarrow[R']{} \!\!\!\!\rightarrow t'$. □

9.5 <u>Definition</u> (continuation of 9.3)

If $n = 1$ and $\langle l_1, r_1 \rangle \in R'$, then $\langle t, t' \rangle$ is called a <u>simple critical pair</u>. $\langle t, t' \rangle$ is <u>including w.r.t. $\langle l_1, \ldots, l_n, f \rangle$</u> if $fl_i = l_i$ for some $1 \leq i \leq n$. <u>CRIT(R,R')</u> and <u>SCRIT(R,R')</u> denote the sets of critical resp. simple critical pairs of $\langle R, R' \rangle$. A subset M of CRIT(R,R') is <u>including</u> if M contains an including pair.

The critical pairs considered in Knuth, Bendix /43/ and Huet /30/ are simple ones.

9.6 <u>Example</u>

Let $\langle l, r \rangle \in R$ and $\langle l_1, r_1 \rangle, \langle l_2, r_2 \rangle \in R'$ be given by

$$l = A(B(Cy_1, y_2, Dy_2) \text{ and } r = A(y_1, y_2),$$

$$l_1 = B(y_1, Dy_2) \text{ and } r_1 = B(y_1, y_2),$$

$$l_2 = DDy_2 \text{ and } r_2 = y_2.$$

With $u = A(x_1, x_2)$, $gx_1 = B(Cy_1, y_2)$ and $gx_2 = Dy_2$ we obtain 9.3 (i) and (ii): Clearly, $l = gu$. Let $f \in Z(T(A))$ and $\beta : X \longrightarrow X$ satisfy

$$\beta y_i = z_i \notin \{y_1, y_2\},\ i = 1,2,$$

$$f y_1 = C z_1,\ f z_2 = D y_2$$

and

$$f x = x \text{ for all } x \in X - \{y_1, z_2\}.$$

Then $\langle f,\beta\rangle$ is a most general unifier of $\langle\langle g x_1, g x_2\rangle,\langle l_1, l_2\rangle\rangle$ because

$$f\beta g x_1 = f\beta B(C y_1, y_2) = f B(C z_1, z_2)$$
$$= B(C z_1, D y_2) = f B(y_1, D y_2) = f l_1$$

and

$$f\beta g x_2 = f\beta D y_2 = f D z_2 = D D y_2$$
$$= f D D y_2 = f l_2.$$

Since

$$f\beta r = f A(z_1, z_2) = A(z_1, D y_2)$$

and

$$f h u = f A(r_1, r_2) = f A(B(y_1, y_2), y_2)$$
$$= A(B(C z_1, y_2), y_2)$$

with h defined as in 9.3,

$$\langle t, t'\rangle = \langle A(z_1, D y_2), A(B(C z_1, y_2), y_2)\rangle$$

is a critical pair of $\langle R, R'\rangle$. $\langle t,t'\rangle$ is including w.r.t. $\langle l_1, l_2, f\rangle$ because $f l_1 = B(C z_1, D y_2) \neq l_1$. □

For proof purposes the R-reduction relation on T(A) (cf. 4.2) has to be analyzed in detail. Each reduction step can be decomposed as follows:

9.7 Reduction-analysis Lemma

Let $A \in K$, $R \subseteq T(A)^2$ and $t \xrightarrow[R]{} t'$. Then there are $\langle l,r \rangle \in R$, $f,f',g \in Z(T(A))$ and a term $u \in T(A)$, which is linear in some $x \in X$, such that $var(u) - var(t) = \{x\}$ $fu = t$, $f'u = t'$, $fx = gl$, $f'x = gr$ and $fy = f'y = y$ for all $y \in var(t)$.

Proof by induction on the definition of $\xrightarrow[R]{}$ (cf. 4.2):

(i) If $\langle t,t' \rangle \in R$, then the lemma follows for $\langle l,r \rangle = \langle t,t' \rangle$, $u = x \in X - var(t)$, $g = inc$ and $f,f' \in Z(T(A))$ with $fx = l$, $f'x = r$ and $fy = f'y = y$ for all $y \in var(t)$.

(ii) Let $t = \sigma(t_1,\ldots,t_n)$, $t' = \sigma(t'_1,\ldots,t'_n)$, $t_i \xrightarrow[R]{} t'_i$ for some $1 \leq i \leq n$, and $t_j = t'_j$ for all $1 \leq j \leq n$ with $j \neq i$. By induction hypothesis, there are $\langle l,r \rangle \in R$, $f,f',g \in Z(T(A))$ and a term $u_i \in T(A)$, which is linear in some $x \in X$, such that $var(u_i) - var(t_i) = \{x\}$, $fu_i = t_i$, $f'u_i = t'_i$, $fx = gl$, $f'x = gr$ and $fy = f'y = y$ for all $y \in var(t_i)$. In addition, we may assume w.l.o.g. that $x \notin var(t)$ and thus $fy = f'y = y$ for all $y \in var(t)$. With $u = \sigma(t_1,\ldots,t_{i-1},u_i,t_{i+1},\ldots,t_n)$ we conclude the lemma.

(iii) Let $t = hv$ and $t' = hv'$ for some $h \in Z(T(A))$ and $v,v' \in T(A)$ with $v \xrightarrow[R]{} v'$. By induction hypothesis, there are $\langle l,r \rangle \in R$, $f_o,f'_o,g_o \in Z(T(A))$ and a term $u_o \in T(A)$, which is

linear in some $x \in X$, such that $var(u_o) - var(v) = \{x\}$, $f_o u_o = v$, $f'_o u_o = v'$, $f_o x = g_o l$, $f'_o x = g_o r$ and $f_o y = f'_o y = y$ for all y $var(v)$.
In addition, we may assume w.l.o.g. that

$$x \notin var(t) \tag{*}$$

and thus $fy = f'y = y$ for all $y \in var(t)$.
Let $f, f', g'_o \in Z(T(A))$ be defined by

$$fy = \begin{cases} hf_o y & \text{if } y = x \\ y & \text{otherwise} \end{cases} \qquad f'y = \begin{cases} hf'_o y & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

and

$$g'_o y = \begin{cases} hy & \text{if } y \in var(u_o) \cap var(v) \\ y & \text{otherwise.} \end{cases}$$

With $u = g'_o u_o$ we conclude the conjecture as follows:

First we observe that
$var(u) \subseteq var(h(var(v))) \cup (var(u_o) - var(v)) \subseteq var(t) \cup \{x\}$.
Hence $var(u) - var(t) \subseteq \{x\}$. By definiton of g'_o, $x \in var(u_o) - var(v)$ implies $x \in var(u)$. Thus by (*), $x \in var(u) - var(t)$. Therefore, $\underline{var(u) - var(t) = \{x\}}$.
Since u_o is linear in x, (*) implies that u is also linear in x. Moreover,

a) for all $y \in var(u_o) \cap var(v)$ $fg'_o y = fhy = hy = hf_o y$ and $f'g'_o y = f'hy = hy = hf'_o y$ because $hy \subseteq t$ and $x \notin var(t)$,

b) $fg_o x = fx = hf_o x$ and $f'g_o x = f'x = hf'_o x$.

Since $x = var(u_o)-var(v)$, a) and b) imply $fg_o u_o = hf_o u_o$ and $f'g_o u_o = hf'_o u_o$. Therefore,

$$fu = fg'_o u_o = hf_o u_o = hv = t,$$

$$f'u = f'g'_o u_o = hf'_o u_o = hv' = t',$$

$$fx = hf_o x = hg_o l, \; f'x = hf'_o x = hg_o r$$

and

$$fy = f'y = y \text{ for all } y \in var(t). \quad \square$$

9.8 <u>Definition</u>

Let $A \in K$, $t \in T(A)$ and $x \in X$. $w \in \mathbb{N}^*$ is an <u>occurrence of x in t</u> if either

$$w = \varepsilon \text{ and } t = x$$

or

$t = \sigma t'$ and $w = iw'$ for some $\sigma \in OP$, $t' \in T(A)^*$, $1 \leq i \leq rank(\sigma)$ and some occurrence w' of x in t'_i.

If t is linear, we write <u>occ(x,t)</u> for the unique occurrence of x in t. The <u>prefix order $\leq$</u> on $\mathbb{N}^*$ is defined by:

$$w \leq w' \text{ iff } ww'' = w' \text{ for some } w'' \in \mathbb{N}^*.$$

The following two lemmata are crucial for Thm. 9.15, which characterizes absolute confluence by a critical pair condition.

9.9 Independent-reductions Lemma

Let $A \in K$, $R \subseteq T(A)^2$, $\langle l', r' \rangle \in T(A)^2$, $h \in Z(T(A))$ and $hl' = t \xrightarrow[R]{} t'$. Let $\langle l, r \rangle$, f, f', g, u, x be as in Lemma 9.7, and assume that there is $z \in var(l')$ with $w \leq occ(x,u)$ for some occurrence w of z in l'.

Then there is $h' \in Z(G(A))$ such that $t' \xrightarrow[R]{\Delta} h'l'$ and $hy \xrightarrow[R]{\Delta} h'y$ for all $y \in var(l')$.

Proof:

Since $t \in G(A)$, $var(t)$ is empty.
There are l'', $r'' \in T(A)$ and $\alpha : X \twoheadrightarrow X$ such that l'' is linear, $\alpha l'' = l'$, $\alpha r'' = r'$ and $\alpha^{-1}(z) \subseteq var(l'')$.
Hence $w = occ(z', l'')$ for some $z' \in \alpha^{-1}(z)$. Since

$$h\alpha l'' = hl' = t = fu \text{ and } w \leq occ(x,u),$$

there is a subterm u' of u such that

$$h\alpha z' = fu' \text{ and } w \cdot occ(x,u') = occ(x,u).$$

Let $\gamma \in Z(T(A))$ be defined by

$$\gamma y = \begin{cases} u' & \text{if } y = z' \\ h\alpha y & \text{otherwise.} \end{cases}$$

Thus $f\gamma z' = fu' = h\alpha z' = fh\alpha z'$ and $f\gamma y = fh\alpha y$ for all $y \in X - \{z'\}$ because $h\alpha z'$ is a subterm of t (cf. 9.7). Therefore $f\gamma = fh\alpha$, and we obtain

$$f\gamma l'' = fh\alpha l'' = h\alpha l'' = hl' = t = fu. \qquad (1)$$

Moreover, $\mathrm{var}(\gamma l'') = \mathrm{var}(u') \subseteq \mathrm{var}(u) = \{x\} \subseteq \mathrm{var}(u')$, and thus

$$\mathrm{var}(\gamma l'') = \mathrm{var}(u') = \{x\}. \tag{2}$$

By definition of γ,

$$\begin{aligned} \mathrm{occ}(x,u) &= \mathrm{occ}(z',l'')\cdot\mathrm{occ}(x,u') \\ &= \mathrm{occ}(z',l'')\cdot\mathrm{occ}(x,\gamma z') = \mathrm{occ}(x,\gamma l''). \end{aligned} \tag{3}$$

(1) - (3) imply $\gamma l'' = u$. Define $h_1, h_2 \in Z(T(A))$ by

$$h_1 y = \begin{cases} f'\gamma y & \text{if } y \in \mathrm{var}(l'') \\ h\alpha y & \text{otherwise} \end{cases}$$

and

$$h_2 y = \begin{cases} f'u' & \text{if } y \in \alpha^{-1}(z) \\ h_1 y & \text{otherwise.} \end{cases}$$

One concludes from $fx \xrightarrow[R]{} f'x$ and (2) that for all $y \in \mathrm{var}(l'')$ $f\gamma y \xrightarrow[R]{\Delta} f'\gamma y$, i.e.

$$h\alpha = f\gamma \xrightarrow[R]{\Delta} h_1. \tag{4}$$

Furthermore, $h_1 z' = f'\gamma z' = f'u'$; and $\alpha^{-1}(z) \subseteq \mathrm{var}(l'')$ and $hz \leq t$ imply

$$h_1 y = f'\gamma y = f'h\alpha y = h\alpha y = hz = h\alpha z' = fu'$$

for all $y \in \alpha^{-1}(z) - \{z'\}$. Hence for all $y \in \alpha^{-1}(z)$

$$h_1 y \in \{fu', f'u'\}. \tag{5}$$

$fx \xrightarrow[R]{} f'x$ and (2) result in $fu' \xrightarrow[R]{\Delta} f'u'$ so that by (5),

$$h_1 \xrightarrow[R]{\Delta} h_2. \qquad (6)$$

Since for all $y, y' \in X$ $\alpha y = \alpha y'$ implies $h_2 y = h_2 y'$, we obtain $h'\alpha = h_2$ for some $h' \in Z(T(A))$. Thus (6) yields

$$t' = f'u = f'\gamma l'' = h_1 l'' \xrightarrow[R]{\Delta} h_2 l''$$
$$= h'\alpha l'' = h'l' .$$

Moreover,

$$h\alpha z' = fu' \xrightarrow[R]{\Delta} f'u' = h_2 z' = h'\alpha z',$$
$$h\alpha y = h_1 y \xrightarrow[R]{\Delta} h_2 y = h'\alpha y$$

for all $y \in var(\alpha^{-1} z) - \{z'\}$, and by (4),

$$h\alpha y = fh\alpha y \xrightarrow[R]{\Delta} h_1 y = h_2 y = h'\alpha y$$

for all $y \in var(l'') - var(\alpha^{-1} z)$.
Hence for all $y \in var(l'')$

$$h\alpha y \xrightarrow[R]{} h'\alpha y. \qquad (7)$$

One concludes from $l' = \alpha l''$ that each $y \in var(l')$ satisfies $y = \alpha z'$ for some $z' \in var(l'')$. Hence (7) amounts to

$$hy \xrightarrow[R]{\Delta} h'y \text{ for all } y \in var(l'). \quad \square$$

9.10 Dependent-reductions Lemma

Let $A \in K$, $R, R' \subseteq T(A)^2$, $<l', r'> \in R'$, $h \in Z(T(A))$ and $hl' = t \xrightarrow[R]{} t'$ such that for all $<l, r> \in R$ $var(r) \subseteq var(l)$.
Let $<l,r>, f, f', g, u, x$ be as in Lemma 9.7, and assume that all occurrences w of variables in l' satisfy $w \not\leq occ(x,u)$.

Then there are a simple critical pair $\langle v,v'\rangle$ of $\langle R',R\rangle$ and $\gamma \in Z(T(A))$ such that $\gamma\varphi l' = t$ and $\gamma v' = t'$ where $\langle\langle l',r'\rangle,\varphi\rangle$ is a generator of $\langle v,v'\rangle$.

Proof:

Since $fu = t = hl'$ and $w \not\leq occ(x,u)$ for all occurrences w of variables in l', one obtains $\alpha \in Z(T(A))$ with $\alpha u = l'$ and $x \in X$. Thus

$$\{y \in var(u) / \alpha y \notin X\} = \{x\}$$

because $var(u) = \{x\}$. We choose an injective mapping $\beta: X \longrightarrow X$ with $var(\beta l') \cap var(l) = \emptyset$ and define $h_o \in Z(T(A))$ by

$$h_o y = \begin{cases} gy & \text{if } y \in var(l) \\ h\beta^{-1} y & \text{if } y \in var(\beta l') \\ y & \text{otherwise.} \end{cases}$$

$fu = h\alpha u$ implies $fx = h\alpha x$ and thus

$$h_o \beta\alpha x = h\alpha x = fx = gl = h_o l$$

because $\alpha x \leq l'$. Hence $\langle h_o, \beta\rangle$ is a unifier of $\langle \alpha x, l\rangle$. We form a most general unifier $\langle h_1, \beta\rangle$ of $\langle \alpha x, l\rangle$ and define $h' \in Z(T(A))$ by

$$h'y = \begin{cases} r & \text{if } y = x \\ \beta\alpha y & \text{otherwise.} \end{cases}$$

Then $\langle v,v'\rangle = \langle h_1 \beta r', h_1 h' u\rangle$ is a simple critical pair of $\langle R',R\rangle$ with generator $\langle\langle l',r'\rangle, h_1\beta\rangle$.

Since $\langle h_1, \beta\rangle$ is a most general unifier of $\langle \alpha x, l\rangle$, there is $\gamma \in Z(T(A))$ with $\gamma h_1 = h_o$. Hence for all $y \in var(l')$ $\gamma h_1 \beta y = h_o \beta y = hy$, and thus

$$\gamma h_1 \beta l' = hl' = t.$$

Every $y \in var(l)$ satisfies $\gamma h_1 y = h_o y = gy$, which amounts to

$$\gamma h_1 h' x = \gamma h_1 r = gr = f' x. \qquad (1)$$

Since $var(u) = \{x\}$, (1) implies

$$\gamma v' = \gamma h_1 h' u = f' u = t' . \ \Box$$

The last technical lemma in this chapter concerns the relationship between two unifiable terms t,t' and their "greatest common divisor" u:

9.11 Common-divisor Lemma

Let $A \in K$, $t,t' \in T(A)$ and $f,f' \in Z(T(A))$ such that $ft = f't'$, t,t' are linear, $var(t) \cup var(t') \neq \emptyset$ and $var(t) \cap var(t') = \emptyset$. Then there are a linear term u and $g,g' \in Z(T(A))$ such that $gu = t$, $g'u = t'$, $gx = x$ for all $x \in var(t)$, $g'x = x$ for all $x \in var(t')$, and $var(u) = V_1 \cup V_2$ where

$$V_1 = \{x \in var(t) / occ(y,t') \nleq occ(x,t) \text{ for all } y \in var(t')\}$$

and

$$V_2 = \{x \in var(t') / occ(y,t) \nleq occ(x,t') \text{ for all } y \in var(t)\}.$$

Proof: by induction on size(t) + size(t'):

Since $ft = ft'$ and $var(t) \cup var(t') \neq \emptyset$, we have the following three cases.

(i) $t \in X$. We set $u = t$ and define $g,g' \in Z(T(A))$ by

$$gx = \begin{cases} t & \text{if } x = u \\ x & \text{otherwise} \end{cases} \quad \text{and} \quad g'x = \begin{cases} t' & \text{if } x = u \\ x & \text{otherwise} \end{cases}$$

The conjecture follows immediately. (Note that $V_1 = \{t\}$ and $V_2 = \emptyset$.)

(ii) $t' \in X$. We proceed as in case (i).

(iii) $size(t) > 1$ and $size(t') > 1$. Then there are $\sigma \in OP$ and $v, v' \in T(A)^{rank(\sigma)}$ such that $\sigma v = t$, $\sigma v' = t'$ and $fv_i = f'v'_i$ for all $1 \leq i \leq n: = rank(\sigma)$.
By induction hypothesis, for all $1 \leq i \leq n$ with $var(v_i) \cup var(v'_i) \neq \emptyset$ there are linear terms u_i and $g_i, g'_i \in Z(T(A))$ such that $g_i u_i = v_i$, $g'_i u_i = v'_i$, $g_i x = x$ for all $x \in var(v_i)$, $g'_i x = x$ for all $x \in var(v'_i)$, and $var(u_i) = V_{1i} \cup V_{2i}$ where

$$V_{1i} = \{x \in var(v_i) / occ(y, v'_i) \nleq occ(x, v_i) \text{ for all } y \in var(v'_i)\}$$

and

$$V_{2i} = \{x \in var(v'_i) / occ(y, v_i) \nleq occ(x, v'_i) \text{ for all } y \in var(v_i)\}.$$

For all $1 \leq i \leq n$, let

$$u'_i = \begin{cases} u_i & \text{if } var(v_i) \cup var(v'_i) \neq \emptyset \\ v_i & \text{otherwise.} \end{cases}$$

The conjecture is valid for $u = \sigma(u'_1, \ldots, u'_n)$ and $g, g' \in Z(T(A))$ defined by

$$gx = \begin{cases} g_i x & \text{if } x \in var(u_i) \text{ for some } i \\ x & \text{otherwise} \end{cases}$$

and

$$g'x = \begin{cases} g'_i x & \text{if } x \in var(u_i) \text{ for some } i \\ x & \text{otherwise.} \end{cases}$$

(g and g' are well-defined because t,t' are linear and $var(u_i) \subseteq var(t_i) \cup var(t'_i)$ for all i.)
We get $gu = \sigma v = t$, $g'u = \sigma v' = t'$, $gx = x$ for all $x \in var(t) = \bigcup_{i=1}^{n} var(v_i)$, and $g'x = x$ for all $x \in var(t') = \bigcup_{i=1}^{n} var(v'_i)$.

Furthermore, let $x \in V_1$. Then there is $1 \leq i \leq n$ such that $x \in var(v_i)$ and $occ(y, v'_i) \not\leq occ(x, v_i)$ for all $y \in var(v'_i)$. Hence $x \in V_{1i}$. Vice versa, suppose that $x \in V_{1i}$ for some $1 \leq i \leq n$. Then $occ(y,t') \not\leq occ(x,t)$ for all $y \in var(t')$, i.e. $x \in V_1$. Therefore, $V_1 = \bigcup_{i=1}^{n} V_{1i}$ and analogously, $V_2 = \bigcup_{i=1}^{n} V_{2i}$. Finally,

$$var(u) = \bigcup_{i=1}^{n} var(u'_i) = \bigcup_{i=1}^{n} (V_{1i} \cup V_{2i})$$
$$= V_1 \cup V_2. \quad \square$$

9.12 Definition

Let $A \in K$ and $R \subseteq T(A)^2$. $\langle t,t' \rangle \in T(A)^2$ is <u>absolutely $\langle R,A \rangle$-unifiable</u> if for all $f \in Z(G(A))$

$$fX \subseteq NF(R) \quad \text{implies} \quad \begin{array}{ccc} ft & \xrightarrow[R]{*} & u \\ ft' & \xrightarrow[R]{*} & u \end{array}$$

for some $u \in G(A)$.

$M \subseteq T(A)^2$ is absolutely $\langle R,A\rangle$-unifiable if all $\langle t,t'\rangle \in M$ are absolutely $\langle R,A\rangle$-unifiable. $\langle t,t'\rangle \in T^2$ (resp. $M \subseteq T^2$) is absolutely R-unifiable if $\langle t,t'\rangle$ (resp. M) is absolutely $\langle R,A\rangle$-unifiable for all $A \in K$.

9.13 Definition

Let $A \in K$ and $R \subseteq T(A)^2$. $\langle R,A\rangle$ is weakly confluent if for all $t,u,u' \in G(A)$

$$\begin{tikzcd}[row sep=small] & u & & u \arrow[dr, "R"', "*"] & \\ t \arrow[ur, "R"] \arrow[dr, "R"'] & & \text{implies} & & t' \\ & u' & & u' \arrow[ur, "R", "*"'] & \end{tikzcd}$$

for some $t' \in G(A)$.

9.14 Proposition (Huet /30/, Lemma 2.4)

Let $A \in K$ and R be a normalizing relation on T(A). $\langle R,A\rangle$ is weakly confluent iff $\langle R,A\rangle$ is absolutely confluent.

Proof:
Suppose that

$$\begin{tikzcd}[row sep=small] & u \\ t \arrow[ur, "R", "*"'] \arrow[dr, "R"', "*"] & \\ & u' \end{tikzcd}$$

for some t,u,u'$\in$G(A). By induction on t w.r.t. $\longrightarrow \cap G(A)^2$ one obtains t'$\in$G(A) with

$$u \xrightarrow[R]{*} t' \xleftarrow[R]{*} u' \;. \quad \square$$

9.15 <u>Absolute-confluence Theorem</u>

Let A$\in$K and R be a normalizing term rewriting system on T(A). <R,A> is absolutely confluent iff SCRIT(R,R) is absolutely <R,A>-unifiable (cf. Def. 9.5).

<u>Proof:</u>

Let <R,A> be absolutely confluent and <t,t'>$\in$SCRIT(R,R). By Prop. 9.4, there is $t_o \in$T(A) with

$$t \xleftarrow[R]{*} t_o \xrightarrow[R]{*} t' \;.$$

Hence for all f$\in$Z(G(A))

$$ft \xleftarrow[R]{*} ft_o \xrightarrow[R]{*} ft' \quad \text{and thus} \quad ft \xrightarrow[R]{*} u \xleftarrow[R]{*} ft'$$

for some $u \in G(A)$. Therefore, SCRIT(R,R) is absolutely $\langle R,A\rangle$-unifiable.

Let SCRIT(R,R) be absolutely $\langle R,A\rangle$-unifiable. By Prop. 9.14, absolute confluence of $\langle R,A\rangle$ follows from weak confluence. So let

$$\begin{tikzcd} & t_1 \\ t \arrow[ur, "R"] \arrow[dr, "R"'] & \\ & t_2 \end{tikzcd}$$

for some $t, t_1, t_2 \in G(A)$. By Lemma 9.7, for $i = 1,2$ there are $\langle l_i, r_i\rangle \in R$, $f_i, f'_i, g_i \in Z(T(A))$ and a linear term u_i such that $var(u_i) = \{x_i\}$ for some x_i, $f_i u_i = t$, $f'_i u_i = t_i$, $f_i x_i = g_i l_i$ and $f'_i x_i = g_i r_i$.
W.l.o.g. let $x_1 \neq x_2$. We define

$$V_1 = \{x \in \{x_1\} \,/\, occ(x_2,u_2) \not< occ(x,u_1)\}$$

and

$$V_2 = \{x \in \{x_2\} \,/\, occ(x_1,u_1) \not\leq occ(x,u_2)\}.$$

Since $f_1 u_1 = f_2 u_2$, Lemma 9.11 implies that there are a linear term u and $h_1, h_2 \in Z(T(A))$ with $h_1 u = u_1$, $h_2 u = u_2$, $h_1 x_1 = x_1$, $h_2 x_2 = x_2$ and $var(u) = V_1 \cup V_2$.
Therefore, $f_1 h_1 u = f_1 u_1 = f_2 u_2 = f_2 h_2 u$ and thus

$$f_1 h_1 x = f_2 h_2 x \qquad (1)$$

for all $x \in var(u)$. We define $f \in Z(T(A))$ by

$$fx = \begin{cases} f_1 x & \text{if } x \in V_1 \\ f_2 x & \text{if } x \in V_2 \\ x & \text{otherwise.} \end{cases}$$

Since $V_1 \subseteq var(u)$ and $h_2u = u_2$, we have

$$var(h_2x) \subseteq var(u_2) = \{x_2\}$$

for all $x \in V_1$. This gives $f_2y \xrightarrow[R]{} f'_2y$ for all $y \in var(h_2x)$.
Thus by (1),

$$fx = f_1x = f_1h_1x = f_2h_2x \xrightarrow[R]{\Delta} f'_2h_2x \tag{2}$$

for all $x \in V_1$. Analogously, for all $x \in V_2$

$$fx \xrightarrow[R]{\Delta} f'_1h_1x.$$

Suppose that each $x \in var(u)$ satisfies

$$\begin{tikzcd}[column sep=small] f'_1h_1x \arrow[dr, "R", "*"'] & \\ & t_x \\ f'_2h_2x \arrow[ur, "*", "R"'] & \end{tikzcd} \tag{3}$$

for some $t_x \in G(A)$. Then

$$\begin{tikzcd}[column sep=small] t_1 = f'_1u_1 = f'_1h_1u \arrow[dr, "R", "*"'] & \\ & u' \\ t_2 = f'_2u_2 = f'_2h_2u \arrow[ur, "*", "R"'] & \end{tikzcd}$$

for some $u' \in G(A)$, and $\langle R, A\rangle$ will be weakly confluent. Hence it remains to show (3).
Let $x \in V_1$. If $fx = f'_2h_2x$, then

$$f'_2h_2x = fx = f_1x = f_1x_1 = g_1l_1 \xrightarrow[R]{} g_1r_1 = f'_1x_1 = f'_1x = f'_1h_1x.$$

Choosing $t_x = f'_1h_1x$ we conclude (3) from

$$f'_1h_1x = f'_1x \subseteq f'_1u_1 = t_1 \in G(A).$$

Let $fx \neq f'_2h_2x$. Then by (2),

$$fx \xrightarrow[R]{} f'_2h_2x.$$

By Lemma 9.7,
there are $\langle l,r\rangle \in R$, $h,h' \in Z(T(A))$ and a linear term u' such that $var(u') = \{z\}$ for some $z \in X$, $hu' = fx$ and $h'u' = f'_2h_2x$. We have two cases:

(i) $w \leq occ(z,u')$ for some occurrence w of a variable in l_1.

(ii) $w \not\leq occ(z,u')$ for all occurrences w of variables in l_1.

Let (i) hold. By Lemma 9.9, $g_1l_1 = fx \xrightarrow[R]{} f_2'h_2x$ implies $f'_2h_2x \xrightarrow[R]{\Delta} g'_1l_1$ for some $g'_1 \in Z(G(A))$, and $g_1y \xrightarrow[R]{} g'_1y$ for all $y \in var(l_1)$. We set $t_x = g'_1r_1$ and conclude (3) from

$$f'_1h_1x = f'_1x = g_1r_1 \xrightarrow[R]{*} g'_1r_1$$

because $var(r_1) \subseteq var(l_1)$.
Let (ii) hold. By Lemma 9.10, $g_1l_1 = fx \xrightarrow[R]{} f_2'h_2x$ implies $\gamma\varphi l_1 = fx$ and $\gamma v' = f'_2h_2x$ for some $\gamma \in Z(G(A))$ and a simple critical pair $\langle v,v'\rangle$ of $\langle R,R\rangle$ with generator $\langle\langle l_1,r_1\rangle,\varphi\rangle$. Since R is normalizing, some $g \in Z(G(A))$ satisfies

$$\gamma y \xrightarrow[R]{*} gy \in NF(R)$$

for all $y \in X$.

$\gamma\varphi l_1 = fx = g_1 l_1$ and $var(r_1) \subseteq var(l_1)$ imply $\gamma v = \gamma\varphi r_1 = g_1 r_1$. Since $\langle v,v'\rangle$ is absolutely $\langle R,A\rangle$-unifiable, we obtain

$$\begin{tikzcd}
f'_1 h_1 x = f'_1 x = g_1 r_1 = \gamma v \arrow[r, "R", "*"'] & gv \arrow[dr, "R *"] & \\
& & t_x \\
f'_2 h_2 x = \gamma v' \arrow[r, "R"'] & gv' \arrow[ur, "* R"'] &
\end{tikzcd}$$

for some $t_x \in G(A)$.

Hence (3) holds for all $x \in V_1$. An analogous proof yields (3) for all $x \in V_2$, and thus (3) holds for all $x \in var(u) = V_1 \cup V_2$. □

Thms. 4.19 and 9.15 amount to the following consistency criterion:

9.16 Consistency Theorem

Let either H be a linear, normalizing and base-complete relation on T such that for all $\langle l,r\rangle \in H$ $op(l) \cap POP = \emptyset$, or let $K = \{\emptyset\}$ and $H = \emptyset$. Let R be a normalizing and base-consistent term rewriting system on T that includes $E \cup H$. If SCRIT(R,R) is absolutely R-unifiable, then PAR is consistent w.r.t. $\langle BPAR,K\rangle$ (cf. 2.7). □

9.17 Definition

Given relations R and R' on T, $\langle t,t'\rangle \in T^2$ (resp.

$M \subseteq T^2$) is called <u>absolutely R-convergent w.r.t. R'</u> if (for all $\langle t,t'\rangle \in M$) there is $\langle u,u'\rangle \in R'$ such that

$$t \xrightarrow[R]{*} u \quad \text{and} \quad t' \xrightarrow[R]{*} u' \ .$$

Clearly, if $M \subseteq T^2$ is absolutely R-convergent w.r.t. = (equality on T), then M is absolutely R-unifiable. This gives rise to a "practical" consistency criterion derived from Thm. 9.16:

9.18 <u>Consistency Theorem</u>

Let E-BE be linear, base-total and included in $T_{OP-POP} \times T$. Let $R \subseteq (T-BT) \times T$ be directly decreasing such that $E \subseteq R$. If SCRIT(R,R) is absolutely R-convergent w.r.t. =, then PAR is consistent w.r.t. $\langle$BPAR,K$\rangle$.

<u>Proof:</u>

Set H = E-BE. Then H is linear and by Thm. 7.2(i), base-complete. Since R is directly decreasing and includes H, H is also directly decreasing. Thus by Thms. 6.10, 6.5 and Coroll. 5.4(i), H and R are normalizing. Since $R \subseteq (T-BT) \times T$, Lemma 4.18 implies that R is base-consistent. Hence the conjecture follows from Thm. 9.16. □

9.19 <u>Example</u> (int)

Let BPAR and PAR be as in Ex. 7.9. We apply Thm. 9.18 in order to show that PAR is consistent w.r.t. $\langle$BPAR,$\{\emptyset\}\rangle$. Clearly, E-BE is linear. By Ex. 7.9, E is directly decreasing and base-total. Hence we set R:= E and enter the loop of decision graph I in section 3.2. We have to check SCRIT(R,R) for absolute R-convergence w.r.t. =. The following list contains all simple critical pairs of <R,R>, each one followed by its generator, where $f_\sigma \in Z(T)$, $\sigma = S, P$, is given by

$$f_\sigma z = \begin{cases} \sigma y & \text{if } z = y \\ y & \text{if } z = x \\ z & \text{otherwise.} \end{cases}$$

$\langle S(x+Py),\ x+y \rangle$	$\langle \underline{i4},\ f_P \rangle$
$\langle P(x+Sy),\ x+y \rangle$	$\langle \underline{i5},\ f_S \rangle$
$\langle P(x-Py),\ x-y \rangle$	$\langle \underline{i7},\ f_P \rangle$
$\langle S(x-Sy),\ x-y \rangle$	$\langle \underline{i8},\ f_S \rangle$
$\langle (x \cdot Py)+x,\ x \cdot y \rangle$	$\langle \underline{i10},\ f_P \rangle$
$\langle (x \cdot Sy)-x,\ x \cdot y \rangle$	$\langle \underline{i11},\ f_S \rangle$

R-convergence of the first four pairs is obtained as follows:

$$S(x+Py) \xrightarrow[\{\underline{i5}\}]{} SP(x+y) \xrightarrow[\{\underline{i1}\}]{} x+y,$$
$$P(x+Sy) \xrightarrow[\{\underline{i4}\}]{} PS(x+y) \xrightarrow[\{\underline{i2}\}]{} x+y,$$
$$P(x-Py) \xrightarrow[\{\underline{i8}\}]{} PS(x-y) \xrightarrow[\{\underline{i2}\}]{} x-y,$$
$$S(x-Sy) \xrightarrow[\{\underline{i7}\}]{} SP(x-y) \xrightarrow[\{\underline{i1}\}]{} x-y.$$

R-convergence of the last two pairs can be deduced if the equations

$$(x+y)-y = x \qquad \underline{i12}$$

and

$$(x-y)+y = x \qquad \underline{i13}$$

are added to R:

$$(x \cdot Py)+x \xrightarrow[\{\underline{i11}\}]{} ((x \cdot y)-x)+x \xrightarrow[\{\underline{i13}\}]{} x \cdot y,$$

$$(x \cdot Sy)-x \xrightarrow[\{\underline{i10}\}]{} ((x \cdot y)+x)-x \xrightarrow[\{\underline{i12}\}]{} x \cdot y.$$

Unfortunately, $\underline{i12}$ and $\underline{i13}$ induce a new list of simple critical pairs and corresponding generators where f_S and f_P are defined as above and f_σ, σ=+,-, is given by

$$f_\sigma z = \begin{cases} \sigma(x,y) & \text{if } z = x \\ z & \text{otherwise:} \end{cases}$$

$$\begin{array}{ll}
\langle x-y,\ x-y\rangle & \langle \underline{i12},\ f_+\rangle \\
\langle x,\ S(x+y)-Sy\rangle & \langle \underline{i12},\ f_S\rangle \\
\langle x,\ P((x+Sy)-y)\rangle & \langle \underline{i12},\ f_S\rangle \\
\langle x,\ P(x+y)-Py\rangle & \langle \underline{i12},\ f_P\rangle \\
\langle x,\ S((x+Py)-y)\rangle & \langle \underline{i12},\ f_P\rangle \\
\langle x+y,\ x+y\rangle & \langle \underline{i13},\ f_-\rangle \\
\langle x,\ P(x-y)+Sy\rangle & \langle \underline{i13},\ f_S\rangle \\
\langle x,\ S((x-Sy)+y)\rangle & \langle \underline{i13},\ f_S\rangle \\
\langle x,\ S(x-y)+Py\rangle & \langle \underline{i13},\ f_P\rangle \\
\langle x,\ P((x-Py)+y)\rangle & \langle \underline{i13},\ f_P\rangle
\end{array}$$

Again, R must be extended by additional equations in order to make the new critical pairs R-convergent w.r.t. =, namely:

$$\begin{array}{ll}
Sx+y = S(x+y) & \underline{i14} \\
Px+y = P(x+y) & \underline{i15} \\
Sx-y = S(x-y) & \underline{i16} \\
Px-y = P(x-y) & \underline{i17}
\end{array}$$

E.g., the second and the third critical pair of the new list get R-convergent as follows:

$$\begin{tikzcd}
S(x+y)-Sy \arrow[dr, "\{\underline{i7}\}"] & & & & \\
& P(S(x+y)-y) \arrow[r, "\{\underline{i14}\}"'] & PS((x+y)-y) \arrow[r, "\{\underline{i2}\}"'] & (x+y)-y \arrow[r, "\{\underline{i12}\}"'] & x. \\
P((x+Sy)-y) \arrow[ur, "\{\underline{i4}\}"'] & & & &
\end{tikzcd}$$

Again, the new equations ($\underline{il4}$ - $\underline{il7}$) induce a new list of simple critical pairs and corresponding generators where f_S and f_P are defined as above, while f'_σ, $\sigma = S,P$, is given by

$$f'z = \begin{cases} x & \text{if } z = x \\ z & \text{otherwise:} \end{cases}$$

$\langle S(Px+y),\ x+y\rangle$	$\langle \underline{il4},\ f'_P\rangle$
$\langle S(x+Sy),\ S(Sx+y)\rangle$	$\langle \underline{il4},\ f_S\rangle$
$\langle S(x+Py),\ P(Sx+y)\rangle$	$\langle \underline{il4},\ f_P\rangle$
$\langle P(Sx+y),\ x+y\rangle$	$\langle \underline{il5},\ f'_S\rangle$
$\langle P(x+Sy),\ S(Px+y)\rangle$	$\langle \underline{il5},\ f_S\rangle$
$\langle P(x+Py),\ P(Px+y)\rangle$	$\langle \underline{il5},\ f_P\rangle$
$\langle S(Px-y),\ x-y\rangle$	$\langle \underline{il6},\ f'_P\rangle$
$\langle S(x-Sy),\ P(Sx-y)\rangle$	$\langle \underline{il6},\ f_S\rangle$
$\langle S(x-Py),\ S(Sx-y)\rangle$	$\langle \underline{il6},\ f_P\rangle$
$\langle P(Sx-y),\ x-y\rangle$	$\langle \underline{il7},\ f'_S\rangle$
$\langle P(x-Sy),\ P(Px-y)\rangle$	$\langle \underline{il7},\ f_S\rangle$
$\langle P(x-Py),\ S(Px-y)\rangle$	$\langle \underline{il7},\ f_P\rangle$
$\langle Sx,\ S(x+y)-y\rangle$	$\langle \underline{il2},\ f'_S\rangle$
$\langle Px,\ P(x+y)-y\rangle$	$\langle \underline{il2},\ f'_P\rangle$
$\langle Sx,\ S(x-y)+y\rangle$	$\langle \underline{il3},\ f'_S\rangle$
$\langle Px,\ P(x-y)+y\rangle$	$\langle \underline{il3},\ f'_P\rangle$

All these critical pairs are R-convergent w.r.t. =. E.g., the first four and the thirteenth pair yield

$$S(Px+y) \xrightarrow[\{\underline{il5}\}]{} SP(x+y) \xrightarrow[\{\underline{il}\}]{} x+y,$$

$$\begin{tikzcd} S(x+Sy) \arrow[dr, "\{\underline{i4}\}"] & \\ & SS(x+y), \\ S(Sx+y) \arrow[ur, "\{\underline{i14}\}"'] & \end{tikzcd}$$

$$\begin{tikzcd}[column sep=small]
S(x+Py) \arrow[r, "\{\underline{i5}\}"] & SP(x+y) \arrow[dr, "\{\underline{i1}\}"] & \\
& & x+y \\
P(Sx+y) \arrow[r, "\{\underline{i4}\}"'] & PS(x+y) \arrow[ur, "\{\underline{i2}\}"'] &
\end{tikzcd}$$

and

$$S(x+y)-y \xrightarrow[\{\underline{i16}\}]{} S((x+y)-y) \xrightarrow[\{\underline{i12}\}]{} Sx,$$

respectively.

Therefore, SCRIT(R,R) is absolutely R-convergent w.r.t. = where R = {<u>i1</u>,...,<u>i17</u>}. It is easy to see that R is directly decreasing and contained in (T-BT)xT. Hence by Thm. 9.18, PAR is consistent w.r.t. <BPAR,K>.

10. Relative confluence

In many cases the assumptions of Thm. 9.16 are too restrictive. In particular, E may not be normalizing. If, e.g., BPAR = <entry,set> (cf. Ex. 8.7), equation s2 of set gives rise to a chain of $\xrightarrow[BE]{}$, namely $(t_i)_{i\in\mathbb{N}}$ with t_i = INS(INS(s,x),y) and t_{i+1} = INS(INS(s,y),x) for all $i\in\mathbb{N}$.

In order to tackle this problem Peterson, Stickel /55/, Lankford, Ballantyne /45/ and Huet /30/ have introduced a weaker notion of confluence which depends on some given equivalence relation. Using Huet's version we adapt this approach to our framework and come up with a consistency criterion (Thm. 10.3) which has partly weaker assumptions than Thm. 9.16. Unfortunately, Ex. 10.4 will show that this criterion is not applicable to our favourite example arrayl (cf. 2.2). Later on we introduce relative confluence as an alternative approach to consistency proofs for specifications with non-normalizing equations.

10.1 Definition

Let $A\in K$, $R\subseteq T(A)^2$ and $\sim$ be an equivalence relation on T(A). <R,A> is confluent modulo $\sim$ if for all $t,t'\in T(A)$

$$t \xleftarrow[R]{*} \sim \xrightarrow[R]{*} t' \text{ implies } t \xrightarrow[R]{*} \sim \xleftarrow[R]{*} t' \; .$$

Let $E0\subseteq T(A)^2$. Huet /30/, Lemma 2.7 and Thm. 3.3, yield the following criterion for confluence modulo $\xleftrightarrow[E0]{*}$.

10.2 Theorem

Let $A\in K$, R be a linear term rewriting system on

$T(A)$ and $E0 \subseteq T(A)^2$ such that $\xrightarrow[R]{} \cdot \xleftrightarrow[E0]{*}$ is well-founded and for all $\langle l,r\rangle \in E0$ $var(l) = var(r)$. $\langle R,A\rangle$ is confluent modulo $\xleftrightarrow[E0]{*}$ iff for all $\langle t,t'\rangle \in SCRIT(R, R \cup E0 \cup E0^{-1}) \cup SCRIT(R \cup E0 \cup E0^{-1}, R)$

$$\begin{tikzcd} t \arrow[r, "R", "*"'] & \cdot \arrow[d, leftrightarrow, "E0", "*"'] \\ t' \arrow[r, "*", "R"'] & \cdot \end{tikzcd}$$ □

10.3 Consistency Theorem

Let $R \subseteq E$ be a linear and base-complete term rewriting system on T such that for all $\langle l,r\rangle \in R$ $op(l) \cap POP = \emptyset$, $E-R \subseteq BE$, $\xrightarrow[R]{} \cdot \xleftrightarrow[E-R]{*}$ is well-founded and for all $\langle l,r\rangle \in E-R$ $var(l) = var(r)$. If all $\langle t,t'\rangle \in SCRIT(R, E \cup (E-R)^{-1}) \cup SCRIT(E \cup (E-R)^{-1}, R)$

$$\begin{tikzcd} t \arrow[r, "R", "*"'] & \cdot \arrow[d, leftrightarrow, "E-R", "*"'] \\ t' \arrow[r, "*", "R"'] & \cdot \end{tikzcd} \qquad (*)$$

and for all $A \in K$ $\langle R,A\rangle$ is base-consistent, then PAR is consistent w.r.t. $\langle BPAR,K\rangle$.

Proof:

Let $A \in K$ and $HSPEC(A) = \langle S, OP(A), E\rangle$. First we show that for all $t,t' \in G(A)$

$$t \equiv_{HSPEC(A)} t' \text{ implies } \begin{tikzcd} t \arrow[r, "R", "*"'] & \cdot \arrow[d, leftrightarrow, "E-R", "*"'] \\ t' \arrow[r, "*", "R"'] & \cdot \end{tikzcd} \qquad (**)$$

Let $t \equiv_{HSPEC(A)} t'$. Then there are a least number n and $t_1,\ldots,t_n, u_1,\ldots,u_{n-1}, v_1,\ldots, v_{n-1} \in G(A)$ with $t_1 = t$, $t_n = t'$ and

$$\begin{tikzcd} u_i \arrow[r, leftrightarrow, "E-R", "*"'] \arrow[d, "R\ *"'] & v_i \arrow[d, "*\ R"] \\ t_i & t_{i+1} \end{tikzcd}$$

for all $1 \le i \le n$. We prove by induction on n that

$$\begin{tikzcd} t \arrow[d, "R\ *"'] & t' \arrow[d, "*\ R"] \\ u \arrow[r, leftrightarrow, "*", "E-R"'] & u' \end{tikzcd}$$

for some $u,u' \in G(A)$. $n = 1$ implies $t = t'$.

Let $n>1$. Since R is normalizing and base-complete, there is $F_A: G(A) \longrightarrow BG(A)$ such that for all $\bar{t} \in G(A)$ $\bar{t} \xrightarrow[R]{*} F_A(\bar{t})$. Thus we have

$$\begin{tikzcd} u_1 \arrow[r, leftrightarrow, "E-R", "*"'] \arrow[d, "R\ *"'] & v_1 \arrow[d, "*\ R"] \\ t & t_2 \arrow[d, "*\ R"] \\ & F_A(t_2) \end{tikzcd}$$

By Thm. 10.2, induction hypothesis and base-consistency of $\langle R,A \rangle$, we obtain

$$\begin{tikzcd} t \arrow[d, "R\ *"'] & & t' \arrow[d, "*\ R"] \\ u \arrow[r, leftrightarrow, "*", "E-R"'] & F_A(t_2) \arrow[r, leftrightarrow, "*", "E-R"'] & u' \end{tikzcd}$$

for some $u,u' \in G(A)$.
Hence (**) holds true, and thus F_A satisfies 4.14(i) and (ii) because $\langle R,A\rangle$ is base-consistent and $E\text{-}R \subseteq BE$. By Lemma 4.14, for all $t,t' \in G(A)$

$$t \equiv_{SPEC(A)} t' \text{ implies } F_A(t) \equiv_{BSPEC(A)} F_A(t')$$

which results in consistency of PAR by 4.14(i). □

10.4 <u>Example</u> (array1)

Let BPAR = <entry,array> (cf. 1.5), PAR = <entry,array1> (cf. 2.2) and K be the class of entry-algebras defined in Ex. 1.11. Since for all

$$\langle l,r\rangle \in E1 = \{\underline{b4}, \underline{b5}, \underline{b8}, \underline{b9}, \underline{n5}, \underline{n8}, \underline{n9}, \underline{n11}, \underline{n12}, \underline{e1}, \underline{e2}, \underline{a1}, \underline{a3}, \ldots, \underline{a6}\}$$

$var(l) \neq var(r)$ or $\langle l,r\rangle \notin BE$, and since $\{\underline{a2}\}$ is not normalizing, R must be chosen such that

$$E1 \subseteq R \subseteq E-\{\underline{a2}\}.$$

But then $\langle t,t'\rangle$ with

$$t = PUT(PUT(a,0,y),Sx,z)$$

and

$$t' = IFA(FALSE,PUT(a,Sx,z),PUT(PUT(a,Sx,z),0,y))$$

is a simple critical pair of $\langle (E-R)^{-1},R\rangle$ with generator $\langle \underline{a2}^{-1},f\rangle$ where $fa = a$, $fn = 0$, $fx = y$, $fm = Sx$ and $fy = z$. A simple check of all possible equations of R proves that 10.3 (*) holds if and only if $t \xleftrightarrow[E-R]{*} t'$ or
$t \xleftrightarrow[E-R]{*} t'' := PUT(PUT(a,Sx,z),0,y)$.

W.l.o.g. assume that E-R = {a2}, and let $t \xleftrightarrow[E-R]{*} u$. There is a least number n such that $t_1 = t$, $t_n = u$ and $t_i \xleftrightarrow[E-R]{} t_{i+1}$. Let $b \in X - \{a,x,y,z\}$. We show by induction on n that

$$f_1(f_3f_4)^k b = u$$
or $$(*)$$
$$f_2f_4(f_3f_4)^k b = u$$

for some k<n where

$$f_1 b = PUT(PUT(a,0,y),Sx,z),$$
$$f_2 b = PUT(PUT(a,Sx,z),0,y),$$
$$f_3 b = IFA(EQN(Sx,0),PUT(a,0,y),b),$$
$$f_4 b = IFA(EQN(0,Sx),PUT(a,Sx,z),b)$$

and for all $c \in X - \{b\}$, $f_1 c = \ldots = f_4 c = c$. n = 1 implies t = u, and thus (*) follows from t = fb. Let n>1. By induction hypothesis,

(i) $f_1(f_3f_4)^k b = t_{n-1}$
or
(ii) $f_2f_4(f_3f_4)^k b = u$

for some k<n-1. If $t_{n-1} \xrightarrow[E-R]{} u$, then (i) implies $u = f_2f_4(f_3f_4)^k b$, while (ii) implies $u = f_1f_3f_4(f_3f_4)^k b = f_1(f_3f_4)^{k+1} b$. If $t_{n-1} \xleftarrow[E-R]{} u$, then (i) yields $u = f_2f_4(f_3f_4)^{k-1} b$, while (ii) results in $u = f_1(f_3f_4)^k b$. Hence (*) holds true.
Since $u \in \{t', t''\}$ does not satisfy (*), we conclude that neither $t \xleftrightarrow[E-R]{*} t'$ nor $t \xleftrightarrow[E-R]{*} t''$ hold, and thus - as we have seen above - <t,t'> does not satisfy 10.3(*). Therefore, Thm. 10.3 cannot be applied in order to show that PAR is consistent w.r.t. <BPAR,K>. □

In the following we introduce the notion of rela-

tive confluence (10.7) which differs from confluence modulo $\xleftrightarrow[E-R]{*}$ mainly because it does not regard E-R as a symmetric relation. A "local" criterion for relative confluence - corresponding to Thm. 9.15 with respect to absolute confluence - will be given by Thm. 10.14. Thms. 10.7, 10.15 and 10.16 provide consistency criteria that assume the existence of relatively confluent relations.
The equivalence closure of reductions used by Huet /30/ will be replaced by a parallel-reduction relation:

10.5 Definition

Let R be a binary relation on T.
R is parallel OP-compatible if for all $w \in S^+$, $s \in S$, $\sigma \in OP_{w,s}$, $t,t' \in T_w$ and $1 \leq j \leq lg(w)$ the following condition holds true:

If $\langle t_j, t'_j \rangle \in R$ and $\langle t_i, t'_i \rangle \in R^{\Delta}$ for all $1 \leq i \leq lg(w)$, then $\langle \sigma t, \sigma t' \rangle \in R$.

R is parallel OP-stable if for all $\langle t,t' \rangle \in R$ and $f,g \in Z(T)$

$$\langle fx, gx \rangle \in R^{\Delta} \text{ for all } x \in X \text{ implies } \langle ft, gt' \rangle \in R.$$

$\xRightarrow[R]{}$ denotes the parallel-R-reduction relation on T, i.e. the least parallel OP-compatible and parallel OP-stable relation on T that includes R.
We write $t \xRightarrow[R]{} t'$ instead of $\langle t,t' \rangle \in \xRightarrow[R]{}$.
Let $A \in K$. Replacing OP, T and Z(T) by OP(A), T(A) and Z(T(A)), respectively, we obtain the notions parallel OP(A)-compatible and parallel OP(A)-stable, and $\xRightarrow[R]{}$ stands for the parallel-R-reduction relation on T(A).

10.6 Definition

Let $A \in K$ and $R \subseteq T(A)^2$. The pair $<R,A>$ is *relatively confluent* if for all $t, u, u' \in G(A)$

$$\begin{tikzcd} & u \\ t \arrow[ur, "R", "*"'] \arrow[dr, Rightarrow, "E-R"'] & \\ & u' \end{tikzcd} \quad \text{implies} \quad \begin{tikzcd} u \arrow[r, "R", "*"'] & v \arrow[dr, Rightarrow, "E-R", "\Delta"'] & \\ & & t' \arrow[dl, "*", "R"'] \\ u' \arrow[r, "*", "R"'] & v' & \end{tikzcd}$$

for some $v, v', t' \in G(A)$.

10.7 Consistency Theorem

Let either H be a linear, normalizing and base-complete relation on T such that for all $<l,r> \in H$ $op(l) \cap POP = \emptyset$, or let $K = \{\emptyset\}$ and $H = \emptyset$.
For all $A \in K$ let $R(A)$ be a normalizing term rewriting system on $T(A)$ such that $H \subseteq R(A)$ and $E\text{-}R(A)$ is a term rewriting system contained in BE. If for all $A \in K$ $<R(A),A>$ is absolutely and relatively confluent, base-complete and base-consistent, then PAR is consistent w.r.t. $<BPAR,K>$.

Proof:

Let $A \in K$ and $HSPEC(A) = <S, OP(A), E \cup H>$. Since $R(A)$ is normalizing and $<R(A),A>$ is absolutely confluent and base-complete, there is a function F_A: $G(A) \longrightarrow BG(A)$ which maps every $t \in G(A)$ to its unique normal form w.r.t. $R(A)$ (cf. Prop. 4.16). First we show that for all $t,t' \in G(A)$

$$t \equiv_{HSPEC(A)} t' \text{ implies}$$
$$F_A(t) \equiv_{BSPEC(A)} F_A(t'). \qquad (*)$$

Let $t \equiv_{HSPEC(A)} t'$. Then there are a least number n and $t_1,\dots,t_n,u_1,\dots,u_n \in G(A)$ with $t_1 = t$, $u_n = t'$, $u_i \xrightarrow[R(A)]{*} t_i$ and

(i) $\quad u_i \xrightarrow[R(A)]{} t_{i+1}$

or

(ii) $\quad u_i \xrightarrow[E-R(A)]{} t_{i+1}$

or

(iii) $\quad u_i \xleftarrow[E-R(A)]{} t_{i+1}$

for all $1 \leq i < n$. We prove $F_A(t) \equiv_{BSPEC(A)} F_A(t')$ by induction on n.

$n = 1$ implies $t' \xrightarrow[R(A)]{*} t$ and thus $F_A(t) = F_A(t')$.

Let $n > 1$. By induction hypothesis, $F_A(t_2) \equiv_{BSPEC(A)} F_A(t')$.

If $i = 1$ satisfies (i), then $F_A(t) = F_A(t_2)$ follows from

$$\begin{tikzcd} & u_1 \arrow[dl, "R(A)"', "*"] \arrow[dr, "R(A)"] & \\ t & & t_2 \end{tikzcd}$$

so that $F_A(t) \equiv_{BSPEC(A)} F_A(t')$.

If $i = 1$ satisfies (ii), then

$$\begin{tikzcd} & u_1 \arrow[dl, "R(A)"', "*"] \arrow[dr, "E-R(A)"] & \\ F_A(t) & & t_2 \end{tikzcd}$$

Since <R(A),A> is relatively confluent, we obtain

$$\begin{tikzcd}
F_A(t) \arrow[d, Rightarrow, "E\text{-}R(A)"', "\Delta"] & t_2 \arrow[d, "*"', "R(A)"] \\
{} \arrow[r, "*", "R(A)"'] & t'_2
\end{tikzcd}$$

for some $t'_2 \in G(A)$. Therefore,

$$\begin{tikzcd}
F_A(t) \arrow[d, Rightarrow, "E\text{-}R(A)"', "\Delta"] & F_A(t_2) \\
{} \arrow[r, "*", "R(A)"'] & t'_2 \arrow[u, "*", "R(A)"']
\end{tikzcd} \qquad (1)$$

Since $F_A(t) \in BG(A)$, E-R(A) is a term rewriting system and <R(A),A> is base-consistent, (1) implies $F_A(t) \equiv_{BSPEC(A)} F_A(t_2)$. Thus $F_A(t) \equiv_{BSPEC(A)} F_A(t')$.

If i = 1 satisfies (iii), we have

$$\begin{tikzcd}
 & t_2 \arrow[dl, "E\text{-}R(A)"'] \arrow[dr, "R(A)", "*"'] & \\
u_1 & & F_A(t_2)
\end{tikzcd}$$

Since <R(A),A> is relatively confluent, we obtain

$$\begin{tikzcd}
u_1 \arrow[d, "R(A)"', "*"] & F_A(t_2) \arrow[d, Rightarrow, "\Delta"', "E\text{-}R(A)"] \\
u'_1 & {} \arrow[l, "*"', "R(A)"]
\end{tikzcd}$$

for some $u'_1 \in G(A)$. Therefore,

$$\begin{tikzcd} F_A(t) & & F_A(t_2) \arrow[dd, Rightarrow, "E\text{-}R(A)", "\triangle"'] \\ & & \\ u'_1 \arrow[uu, "R(A)", "*"'] & & {} \arrow[ll, "R(A)", "*"'] \end{tikzcd} \qquad (2)$$

Since $F_A(t) \in BG(A)$, E-R(A) is a term rewriting system and $\langle R(A),A\rangle$ is base-consistent, (2) implies $F_A(t_2) \equiv_{BSPEC(A)} F_A(t)$. Thus $F_A(t) \equiv_{BSPEC(A)} F_A(t')$.

Hence (*) holds true.

<u>Case 1</u>: H is a linear, normalizing and base-complete relation on T such that for all $\langle l,r\rangle \in H$ $op(l) \cap POP = \emptyset$. (*) agrees with condition 4.14(i). 4.14(ii) follows from base-consistency of $\langle R(A),A\rangle$. Thus by Lemma 4.14, PAR is consistent w.r.t. $\langle BPAR,K\rangle$.

<u>Case 2</u>: $K = \{\emptyset\}$ and $H = \emptyset$. Then SPEC(A) = HSPEC(A) so that $t \equiv_{SPEC(A)} t'$ with $t,t' \in BG(A)$ implies $t \equiv_{BSPEC(A)} t'$ by (*) and base-consistency of $\langle R(A),A\rangle$. Therefore, PAR is consistent w.r.t. $\langle BPAR,K\rangle$. □

Let us now prepare the local criterion for relative confluence given by Thm. 10.14. First we need a decomposition lemma for parallel reductions that corresponds to Lemma 9.7 where we analyzed non-parallel reductions. Lemmata 10.9 and 10.10 correspond to 9.9 and 9.10, respectively. Finally, relative R-unifiability of term pairs (Def.10.11) plays a similar role for relative confluence as absolute R-unifiability does for absolute confluence.

10.8 Parallel-reduction-analysis Lemma

Let $A\in K$, $R\subseteq T(A)^2$ and $t \overset{}{\underset{R}{\Longrightarrow}} t'$. Then there are $f,f'\in Z(T(A))$ and a term $u\in T(A)$, which is linear in X-var(t), such that $fu = t$, $f'u = t'$, for all $y\in var(t)$ $fy = f'y = y$ and for all $x\in var(u)-var(t) = \emptyset$ $fx = gl$, $f'x = g'r$ and $g \overset{\Delta}{\underset{R}{\Longrightarrow}} g'$ for some $\langle l,r\rangle\in R$ and $g,g'\in Z(T(A))$.

Proof by induction on the definition of $\underset{R}{\Longrightarrow}$ (cf. 10.5):

(i) If $\langle t,t'\rangle\in R$, then the conjecture follows for $\langle l,r\rangle = \langle t,t'\rangle$, $u = x\in X\text{-}var(t)$, $g = g' = inc$ and $f,f'\in Z(T(A))$ with $fx = l$, $f'x = r$ and $fy = f'y = y$ for all $y\in var(t)$.

(ii) Let $t = \sigma(t_1,\dots,t_n)$, $t' = \sigma(t'_1,\dots,t'_n)$, $t_{ij} \underset{R}{\Longrightarrow} t'_{ij}$ for some $1\le i1,\dots,ik\le n$ and $t_i = t'_i$ for all $1\le i\le n$ with $i\notin\{i1,\dots,ik\}$. By induction hypothesis, for all $1\le j\le k$ there are $f_j,f'_j\in Z(T(A))$ and a term u_j, which is linear in $X\text{-}var(t_{ij})$, such that $f_ju_j = t_{ij}$, $f'_ju_j = t'_{ij}$, for all $y\in var(t_{ij})$ $f_jy = f'_jy = y$ and for all $x\in var(u_j)-var(t_{ij}) \neq \emptyset$ some $\langle l,r\rangle\in R$ and $g,g'\in Z(T(A))$ satisfy $f_jx = gl$, $f'_jx = gr$ and $g \overset{\Delta}{\underset{R}{\Longrightarrow}} g'$.
W.l.o.g. we may assume that for all $1\le j,r\le k$ with $j \neq r$

$$(var(u_j)-var(t_{ij}))\cap((var(u_r)-var(t_{ir}))\cup var(t)) = \emptyset \qquad (1)$$

and thus $f_jy = f'_jy = y$ for all $y\in var(t)$.
Let $f,f'\in Z(T(A))$ be defined by

$$fx = \begin{cases} f_jx & \text{if } x\in var(u_j)-var(t_{ij}) \text{ and } 1\le j\le k \\ x & \text{otherwise} \end{cases}$$

and

$$f'x = \begin{cases} f'_j x & \text{if } x \in var(u_j) - var(t_{ij}) \text{ and } 1 \leq j \leq k \\ x & \text{otherwise.} \end{cases}$$

Let $u'_{ij} = u_j$ for all $1 \leq j \leq k$ and $u'_i = t_i$ for all $1 \leq i \leq n$ with $i \notin \{i1,\dots,ik\}$. With $u = \sigma(u'_1,\dots,u'_n)$ we obtain the conjecture as follows:

By (1) and since for all $1 \leq j \leq k$ u_j is linear in $X-var(t_{ij})$, u is linear in $X-var(t)$. Moreover,

$$fu = \sigma(fu'_1,\dots,fu'_n) = \sigma(t_1,\dots,t_n) = t,$$

$$f'u = \sigma(f'u'_1,\dots,f'u'_n) = \sigma(t'_1,\dots,t'_n) = t'$$

and $fy = f'y = y$ for all $y \in var(t)$. Finally, for all $x \in var(u)-var(t) = \bigcup_{j=1}^{n} (var(u_j)-var(t_{ij})) = \emptyset$ there are $\langle l,r \rangle \in R$ and $g,g' \in Z(T(A))$ with $fx = gl$, $f'x = gr$ and $g \overset{\Delta}{\underset{R}{\Longrightarrow}} g'$.

(iii) Let $t = hv$ and $t' = h'v'$ for some $h,h' \in Z(T(A))$ and $v,v' \in T(A)$ with $v \underset{R}{\Longrightarrow} v'$ and $h \overset{\Delta}{\underset{R}{\Longrightarrow}} h'$. By induction hypothesis, there are $f_o, f'_o \in Z(T(A))$ and a term u_o, which is linear in $X-var(v)$, such that $f_o u_o = v$, $f'_o u_o = v'$, for all $y \in var(v)$ $f_o y = f'_o y = y$ and for all $x \in var(u_o)-var(v) \neq \emptyset$ some $\langle l,r \rangle \in R$ and $g,g' \in Z(T(A))$ satisfy $f_o x = gl$, $f'_o x = g'r$ and $g \overset{\Delta}{\underset{R}{\Longrightarrow}} g'$.

W.l.o.g. we may assume that

$$(var(u_o)-var(v)) \cap var(t) = \emptyset. \tag{2}$$

Let V be the set of all $x \in X$ with $hx \underset{R}{\Longrightarrow} h'x$. Let $x \in V$. By induction hypothesis, there are $f_x, f'_x \in Z(T(A))$ and a term u_x, which is linear in

X-var(hx), such that for all $y \in$ var(hx) $f_x y = f'_x y = y$ and for all $y \in$ var(u_x)-var$(hx) \neq \emptyset$ some $\langle l,r\rangle \in R$ and $g,g' \in Z(T(A))$ satisfy $f_x y = gl$, $f'_x y = gr$ and $g \overset{\Delta}{\underset{R}{\Longrightarrow}} g'$.

W.l.o.g. suppose that for all $x,y \in V$ with $x \neq y$

$$(\mathrm{var}(u_x)-\mathrm{var}(hx)) \cap ((\mathrm{var}(u_y)-\mathrm{var}(hy)) \cup \mathrm{var}(u_o) \cup \mathrm{var}(t)) = \emptyset. \tag{3}$$

Let $f,f' \in Z(T(A))$ be defined by

$$fy = \begin{cases} hf_o y & \text{if } y \in \mathrm{var}(u_o)-\mathrm{var}(v) \\ f_x y & \text{if } y \in \mathrm{var}(u_x)-\mathrm{var}(hx) \text{ and } x \in V \\ y & \text{otherwise} \end{cases}$$

and

$$f'y = \begin{cases} h'f'_o y & \text{if } y \in \mathrm{var}(u_o)-\mathrm{var}(v) \\ f'_x y & \text{if } y \in \mathrm{var}(u_x)-\mathrm{var}(hx) \text{ and } x \in V \\ y & \text{otherwise.} \end{cases}$$

Moreover, define $g_o \in Z(T(A))$ by

$$g_o x = \begin{cases} u_x & \text{if } x \in \mathrm{var}(u_o) \cap \mathrm{var}(v) \cap V \\ hx & \text{if } x \in (\mathrm{var}(u_o) \cap \mathrm{var}(v))-V \\ x & \text{otherwise.} \end{cases}$$

With $u = g_o u_o$ we conclude the conjecture as follows:
The definition of u yields
$\mathrm{var}(u) \subseteq (\bigcup_{x \in V} \mathrm{var}(u_x)) \cup \mathrm{var}(h(\mathrm{var}(v))) \cup (\mathrm{var}(u_o)-\mathrm{var}(v))$
and thus

$$\mathrm{var}(u)-\mathrm{var}(t) \subseteq (\bigcup_{x \in V} (\mathrm{var}(u_x)-\mathrm{var}(hx))) \cup (\mathrm{var}(u_o)-\mathrm{var}(v)). \tag{4}$$

Since u_o is linear in X-var(v) and for all $x \in V$

u_x is linear in X-var(hx), (3) implies that u is linear in X-var(t). Furthermore,

a) for all $x \in V$ and $y \in var(u_x) - var(hx)$
$fy = f_x y$ and $f'y = f'_x y$,

b) for all $x \in V$ and $y \in var(u_x) \cap var(hx)$
$fy = y = f_x y$ and $f'y = y = f'_x y$,

c) for all $x \in (var(u_o) \cap var(v)) - V$ $fg_o x = fhx$
$= hx = hf_o x$ and $f'g_o x = f'hx = hx = h'x$
$= h'f'_o x$,

d) for all $x \in var(u_o) - var(v)$ $fg_o x = fx$
$= hf_o x$ and $f'g_o x = f'x = h'f'_o x$.

a) and b) imply

e) for all x $var(u_o) \cap var(v) \cap V$ $fg_o x = fu_x$
$= f_x u_x = hx = hf_o x$ and $f'g_o x = f'u_x$
$= f'_x u_x = h'x = h'f'_o x$.

c), d) and e) imply

$$fu = fg_o u_o = hf_o u_o = hv = t$$

and

$$f'u = f'g_o u_o = h'f'_o u_o = h'v' = t' .$$

(2) and (3) yield $fy = y = f'y$ for all $y \in var(t)$.

Let $y \in var(u) - var(t)$. By (4), we have either $y \in var(u_x) - var(hx)$ for some $x \in V$ or $y \in var(u_o) - var(v)$. In the first case there are $<l,r> \in R$ and $g, g' \in Z(T(A))$ such that $fy = f_x y = gl$, $f'y = f'_x y$ $= g'r$ and $g \overset{\Delta}{\underset{R}{\Longrightarrow}} g'$.
The second case provides $<l,r> \in R$ and $g, g' \in Z(T(A))$

with $fy = hf_o y = hgl$, $f'y = h'f'_o y = h'g'r$ and $hg \overset{\Delta}{\underset{R}{\Longrightarrow}} hg'$. $\square$

10.9 Independent-parallel-reductions Lemma

Let $A \in K$, $R \subseteq T(A)^2$, $e = \langle l', r' \rangle \in T(A)^2$ be a linear rule, $h \in Z(T(A))$ and $hl' = t \underset{R}{\Longrightarrow} t'$. Let f, f', u be as in Lemma 10.8, and assume that for all $x \in var(u)$ there is $z_x \in var(l')$ with $occ(z_x, l') \leq occ(x,u)$.
Then

$$\begin{tikzcd} hr' \arrow[dr, Rightarrow, "R", "\Delta"'] & \\ & {} \\ t' \arrow[ur, "\{e\}"'] & \end{tikzcd}$$

Proof:

Since $t \in G(A)$, $var(t)$ is empty.
Let $x \in var(u)$. Since $hl' = t = fu$ and $occ(z_x, l') \leq occ(x,u)$, there is a subterm u_x of u such that

$$hz_x = fu_x \text{ and } occ(z_x, l') \cdot occ(x, u_x) = occ(x,u).$$

Let $\gamma \in Z(T(A))$ be defined by

$$\gamma y = \begin{cases} u_x & \text{if } y = z_x \text{ and } x \in var(u) \\ hy & \text{otherwise.} \end{cases}$$

Hence $f\gamma z_x = fu_x = hz_x = fhz_x$ for all $x \in var(u)$ because $hz_x \leq t$, and $f\gamma y = fhy$ for all $y \in X - \{z_x / x \in var(u)\}$.
Therefore $f\gamma = fh$, and we obtain

$$f\gamma l' = fhl' = hl' = t = fu. \qquad (1)$$

Moreover,

$$var(\gamma l') = \bigcup_{x\in var(u)} var(u_x) = var(u), \qquad (2)$$

and for all $x\in var(u)$

$$\begin{aligned} occ(x,u) &= occ(z_x,l')\cdot occ(x,u_x) \\ &= (occ(z_x,l')\cdot occ(x,\gamma x) = occ(x,\gamma l'). \qquad (3) \end{aligned}$$

(1) - (3) imply $\gamma l' = u$. Define $h'\in Z(T(A))$ by

$$h'y = \begin{cases} f'\gamma y & \text{if } y\in var(l') \\ hy & \text{otherwise.} \end{cases}$$

Since for all $x\in var(u)$ $fx \underset{R}{\Longrightarrow} f'x$, (2) implies

$$f\gamma y \overset{\Delta}{\underset{R}{\Longrightarrow}} f'\gamma y$$

for all $y\in var(l')$, and thus $fh = f\gamma \overset{\Delta}{\underset{R}{\Longrightarrow}} h'$.
Finally, $t' = f'u = f'\gamma l' = h'l'$ results in

$$\begin{tikzcd} hr' = fhr' \arrow[dr, Rightarrow, "R", "\Delta"'] & \\ & h'r' \\ t' \arrow[ur, "\{e\}"'] & \end{tikzcd}$$

because $fhl' = hl'$ and $var(r')\subseteq var(l')$. □

10.10 <u>Dependent-parallel-reductions Lemma</u>

Let $A\in K$, $R,R'\subseteq T(A)^2$, $\langle l',r'\rangle\in R'$, $h\in Z(T(A))$ and $hl' = t \underset{R}{\Longrightarrow} t'$ such that for all $\langle l,r\rangle\in R$ $var(r)\subseteq var(l)$, $\langle l',r'\rangle$ is a linear rule and $CRIT(R',R)$ is not including (cf. 9.5). Let f,f',u be as in Lemma 10.8, and assume that some $x\in var(u)$ satisfies $occ(y,l')\nleq occ(x,u)$ for all $y\in var(l')$.
Then there are a critical pair $\langle v,v'\rangle$ of $\langle R',R\rangle$

and $\gamma, \gamma' \in Z(T(A))$ such that $\gamma\varphi l' = t$, $\gamma' v' = t'$ and $\gamma \overset{\Delta}{\underset{R}{\Longrightarrow}} \gamma'$ where $\langle\langle l', r'\rangle, \varphi\rangle$ is a generator of $\langle v, v'\rangle$.

<u>Proof:</u>

Since $t \in G(A)$, var(t) is empty. W.l.o.g. suppose that var(l')∩var(u) is empty, and define

$$V_1 = \{x \in \mathrm{var}(l') / \mathrm{occ}(y,u) \not\leq \mathrm{occ}(x,l') \text{ for all } y \in \mathrm{var}(u)\}$$

and

$$V_2 = \{x \in \mathrm{var}(u) / \mathrm{occ}(y,l') \not\leq \mathrm{occ}(x,u) \text{ for all } y \in \mathrm{var}(l')\}.$$

Since hl' = t = fu, Lemma 9.11 provides a linear term u' and $g, g' \in Z(T(A))$ such that gu' = l', g'u' = u, gx = x for all $x \in \mathrm{var}(l')$, g'x = x for all $x \in \mathrm{var}(u)$, and $\mathrm{var}(u') = V_1 \cup V_2$. Hence gx = x for all $x \in V_1$, and thus

$$V_1 \subseteq \{x \in \mathrm{var}(u') / gx \in X\}. \qquad (1)$$

Let $x \in \mathrm{var}(u')$ and $gx \in X$. If x would belong to V_2, we would get g'x = x and thus occ(gx,gu') = occ(x,g'u'). Therefore occ(gx,l') = occ(x,u) in contradiction to $x \in V_2$. Hence

$$\{x \in \mathrm{var}(u') / gx \in X\} \subseteq V_1. \qquad (2)$$

By (1) and (2),

$$\{x \in \mathrm{var}(u') / gx \notin X\} = V_2. \qquad (3)$$

By assumption, V_2 is not empty, and by Lemma 10.8, for all $x \in V_2$ some $\langle l_x, r_x\rangle \in R$ and $g_x, g'_x \in Z(T(A))$ satisfy $fx = g_x l_x$,

$f'x = g'_x r_x$ and $g_x \xrightarrow[R]{\Delta} g'_x$.

Let $x \in V_2$. We choose an injective mapping $\alpha_x : X \longrightarrow X$ such that for all $y \in V_2$ $y \neq x$ implies

$$var(\alpha_x l_x) \cap (var(\alpha_y l_y) \cup var(l') \cup var(u')) = \emptyset \tag{4}$$

and define $h_o \in Z(T(A))$ by

$$h_o y = \begin{cases} g_x \alpha_x^{-1} y & \text{if } y \in var(\alpha_x l_x) \text{ and } x \in V \\ hy & \text{if } y \in var(l') \\ y & \text{otherwise.} \end{cases}$$

Since $fg'u' = fu = t = hl' = hgu'$, we obtain for all $x \in V_2$

$$h_o \alpha_x l_x = g_x l_x = fx = fg'x = hgx$$

and thus $h_o \alpha_x l_x = h_o gx$ because gx is a subterm of l'. Let $V_2 = \{x1, \ldots, xn\}$, $\bar{t} = \langle gx_1, \ldots, gx_n \rangle$ and $\bar{u} = \langle \alpha_{x1} l_{x1}, \ldots, \alpha_{xn} l_{xn} \rangle$. We have just shown that $\langle h_o, id \rangle$ is a unifier of $\langle \bar{t}, \bar{u} \rangle$. We form a most general unifier $\langle h_1, id \rangle$ of $\langle t, u \rangle$ and define $h' \in Z(T(A))$ by

$$h'x = \begin{cases} \alpha_x r_x & \text{if } x \in V_2 \\ gx & \text{otherwise.} \end{cases}$$

(3) implies that $\langle v, v' \rangle = \langle h_1 r', h_1 h' u' \rangle$ is a critical pair of $\langle R', R \rangle$ with generator $\langle\langle l', r' \rangle, h_1 \rangle$. Since $\langle h_1, id \rangle$ is a most general unifier of $\langle t, u \rangle$, there is $\gamma \in Z(T(A))$ with $\gamma h_1 = h_o$. Hence for all $x \in var(l')$ $\gamma h_1 x = h_o x = hx$, and thus

$$\gamma h_1 l' = hl' = t.$$

By assumption, $\langle v, v' \rangle$ is not including so that for

all $x \in V_2$ $h_1 \alpha_x l_x = \alpha_x l_x$. Therefore
$h_1 \alpha_x r_x = \alpha_x r_x$ because $var(r_x) \subseteq var\ (l_x)$.
Define $\gamma' \in Z(T(A))$ by

$$\gamma' y = \begin{cases} g'_x \alpha_x^{-1} y & \text{if } y \in var(\alpha_x l_x) \text{ and } x \in V_2 \\ f'g'y & \text{if } y \in V_1 \\ \gamma y & \text{otherwise.} \end{cases}$$

One obtains

$$\begin{aligned} \gamma' h_1 h' x &= \gamma' h_1 \alpha_x r_x = \gamma' \alpha_x r_x = g'_x r_x \\ &= f'x = f'g'x \end{aligned} \tag{5}$$

for all $x \in V_2$.
Let $x \in V_1$. By (4), $x \notin var(\alpha_y l_y)$ for all $y \in V_2$.
$x \in var(gy)$ for some $y \in V_2$ would imply $occ(y,u') < occ(x,l')$ because $gu' = l'$ and $gy \notin X$ (cf. (3)).
Since $g'u' = u$ and $g'y = y$, we would obtain $occ(y,u) = occ(y,u') < occ(x,l')$ in contradiction to $x \in V_1$. Hence for all $y \in V_2$

$$x \notin var(\alpha_y l_y)\ var(gy),$$

i.e. $x \notin var(\bar{t}) \cup var(\bar{u})$. Thus we may assume that

$$h_1 x = x, \tag{6}$$

which amounts to

$$\gamma' h_1 h' x = \gamma' h_1 gx = \gamma' h_1 x = \gamma' x = f'g'x. \tag{7}$$

(5) and (7) yield

$$\gamma' v' = \gamma' h_1 h' u' = f'g'u' = f'u' = t' .$$

Finally, we have to show $\gamma \overset{\Delta}{\underset{R}{\Longrightarrow}} \gamma'$.
Since $<v,v'>$ is not including, we have for all $x \in V_2$

$$\gamma\alpha_x 1_x = \gamma h_1 \alpha_x 1_x = h_o \alpha_x 1_x$$

and thus for all $y \in var(\alpha_x 1_x)$

$$= h_o y = g_x \alpha_x^{-1} y \overset{\Delta}{\underset{R}{\Longrightarrow}} g'_x \alpha_x^{-1} y = \gamma' y. \qquad (8)$$

Since $h_o g u' = h_o l' = hl' = t = fu = fg'u'$,
(6) implies for all $x \in V_1$

$$\gamma x = \gamma h_1 x = \gamma h_1 g x = h_o g x = fg'x$$

and thus

$$\gamma x = fg'x \overset{\Delta}{\underset{R}{\Longrightarrow}} f'g'x = \gamma' x \qquad (9)$$

because $fu \underset{R}{\Longrightarrow} f'u$ and $g'x$ is a subterm of u.
Hence by definition of γ', $\gamma \overset{\Delta}{\underset{R}{\Longrightarrow}} \gamma'$ follows from (8) and (9). □

10.11 Definition

Let $A \in K$ and $R \subseteq T(A)^2$. $\langle t,t' \rangle \in T(A)^2$ is <u>relatively $\langle R,A \rangle$-unifiable</u> if there is $u \in T(A)$ such that $t \overset{*}{\underset{R}{\longrightarrow}} \overset{\Delta}{\underset{E-R}{\Longrightarrow}} u$ and $\langle u,t' \rangle$ is absolutely $\langle R,A \rangle$-unifiable (cf. Def. 9.12).
$M \subseteq T(A)^2$ is relatively $\langle R,A \rangle$-unifiable if all $\langle t,t' \rangle \in M$ are relatively $\langle R,A \rangle$-unifiable.
$\langle t,t' \rangle \in T^2$ (resp. $M \subseteq T^2$) is <u>relatively R-unifiable</u> if $\langle t,t' \rangle$ (resp. M) is relatively $\langle R,A \rangle$-unifiable for all $A \in K$.

10.12 Definition

Let $A \in K$ and $R \subseteq T(A)^2$. G(R) denotes the set of all pairs $\langle fl, gr \rangle$ with $\langle l, r \rangle \in R$ and $f, g \in Z(G(A))$ such that $f \xRightarrow[R]{\Delta} g$.

10.13 Proposition

Let $A \in K$ and $R \subseteq T(A)^2$. For all $t, t' \in G(A)$,

$$t \xRightarrow[G(R)]{} t' \text{ iff } t \xRightarrow[R]{} t' .$$

Proof:

Clearly, G(R) is a subset of $\xRightarrow[R]{}$. Hence $\xRightarrow[G(R)]{} \subseteq \xRightarrow[R]{}$. Let $t \xRightarrow[R]{} t'$ for some $t, t' \in G(A)$. By Lemma 10.8, there are $f, f' \in Z(T(A))$ and $u \in T(A)$ such that $fu = t$, $f'u = t'$ and $\langle fx, gx \rangle \in G(R)$ for all $x \in var(u)$. Therefore $t \xRightarrow[G(R)]{} t'$. □

10.14 Relative-confluence Theorem

Let $A \in K$ and R be a linear and normalizing term rewriting system on T(A) such that $\langle R, A \rangle$ is absolutely confluent and E-R is a term rewriting system.
$\langle R, A \rangle$ is relatively confluent if the following conditions hold:

(a) For all $e \in R$ CRIT($\{e\}$, E-R) is relatively $\langle R, A \rangle$-unifiable and not including, or CRIT($\{e\}$,G(E-R)) is relatively $\langle R, A \rangle$-unifiable.

(b) For all $\langle t, t' \rangle \in$ SCRIT(E-R,R) and $f \in Z(G(A))$ with $fX \subseteq NF(R)$ some $g \in Z(G(A))$ satisfies

$$ft' \xrightarrow[R]{*} gl$$

and for all $x \in var(l)$

$$f\varphi x \xrightarrow[R]{*} gx$$

where $\langle\langle l,r\rangle,\varphi\rangle$ is a generator of $\langle t,t'\rangle$.

<u>Proof:</u>

Let (a) and (b) hold true and

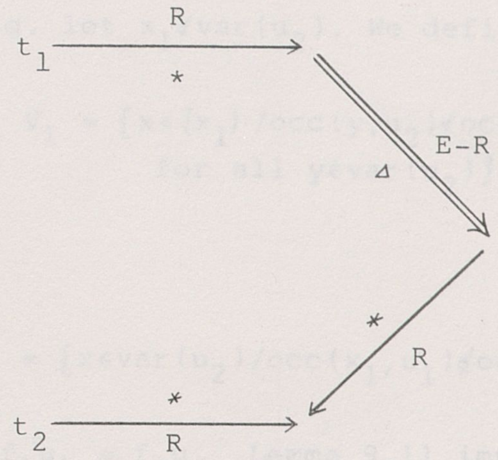
$$\begin{tikzcd} & u \\ t \arrow[ur, "R"', "*"] \arrow[dr, Rightarrow, "E-R"'] & \\ & t_2 \end{tikzcd} \qquad (0)$$

for some $t,u,t_2 \in G(A)$. We prove by induction on t w.r.t. $>_R$ (cf. 4.9) that

$$\begin{tikzcd} u \arrow[r, "R", "*"'] & \cdot \arrow[dr, Rightarrow, "E-R", "\triangle"'] & \\ & & \cdot \arrow[dl, "*"', "R"] \\ t_2 \arrow[r, "*", "R"'] & \cdot & \end{tikzcd} \qquad (1)$$

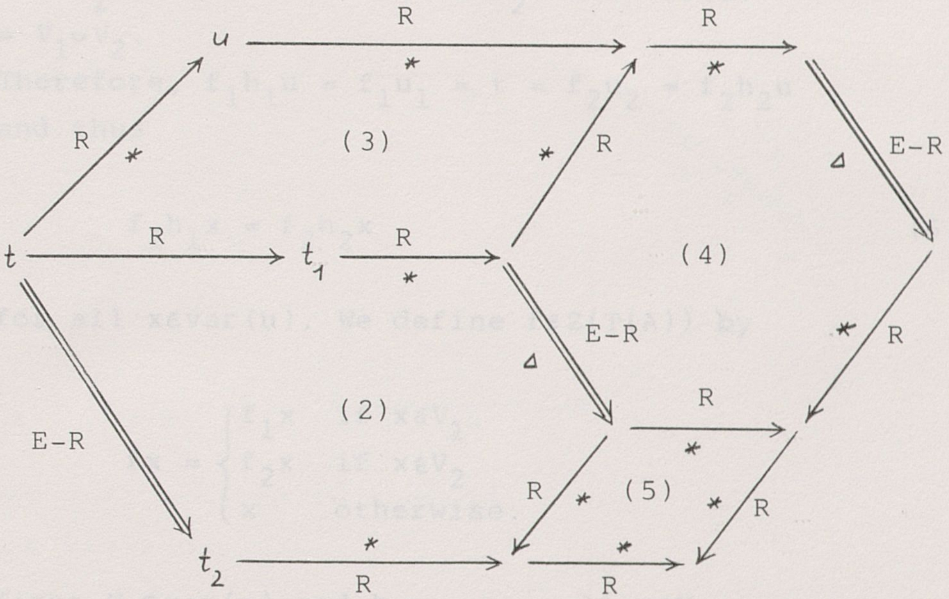
If $t \in NF(R)$, then $t = u$ and thus (1) follows from $u = t \xRightarrow[E-R]{} t_2$. Otherwise $t \xrightarrow[R]{} t_1$ for some $t_1 \in G(A)$.

Provided that



(2)

(1) is obtained as follows:



(3) and (5) result from absolute confluence of $\langle R,A\rangle$. (4) is a consequence of the induction hypothesis. Thus it remains to show (2).

By Lemmata 9.7 and 10.8, (0) implies that there are $\langle l,r\rangle \in R$, $f_1, f'_1, g, f_2, f'_2 \in Z(G(A))$ and linear terms u_1, u_2 such that $var(u_1) = \{x_1\}$ for some $x_1 \in X$,
$f_1u_1 = f_2u_2 = t$, $f'_1u_1 = t_1$,
$f'_2u_2 = t_2$, $f_1x_1 = gl$, $f'_1x_1 = gr$
and for all $x \in var(u_2)$ some $\langle l_x, r_x\rangle \in E\text{-}R$ and $g_x, g'_x \in Z(T(A))$ satisfy $f_2x = g_xl_x$,

$f'_2 x = g'_x r_x$ and $g_x \xRightarrow[E-R]{\Delta} g'_x$.

W.l.o.g. let $x_1 \notin var(u_2)$. We define

$$V_1 = \{x \in \{x_1\} / occ(y,u_2) \nleq occ(x,u_1) \text{ for all } y \in var(u_2)\}$$

and

$$V_2 = \{x \in var(u_2) / occ(x_1,u_1) \nleq occ(x,u_2)\}.$$

Since $f_1 u_1 = f_2 u_2$, Lemma 9.11 implies that there are a linear term u and $h_1, h_2 \in Z(T(A))$ with $h_1 u = u_1$, $h_2 u = u_2$, $h_1 x_1 = x_1$ and $h_2 x = x$ for all $x \in var(u_2)$ and $var(u) = V_1 \cup V_2$.
Therefore, $f_1 h_1 u = f_1 u_1 = t = f_2 u_2 = f_2 h_2 u$ and thus

$$f_1 h_1 x = f_2 h_2 x \tag{6}$$

for all $x \in var(u)$. We define $f \in Z(T(A))$ by

$$fx = \begin{cases} f_1 x & \text{if } x \in V_1 \\ f_2 x & \text{if } x \in V_2 \\ x & \text{otherwise.} \end{cases}$$

Since $V_1 \subseteq var(u)$ and $h_2 u = u_2$, all $x \in V_1$ satisfy $var(h_2 x) \subseteq var(u_2)$. This gives $f_2 y \xRightarrow[E-R]{} f'_2 y$ for all $y \in var(h_2 x)$. Thus by (6),

$$fx = f_1 x = f_1 h_1 x = f_2 h_2 x \xRightarrow[E-R]{\Delta} f'_2 h_2 x \tag{7}$$

for all $x \in V_1$. Analogously, for all $x \in V_2$

$$fx \xrightarrow[R]{\Delta} f'_1 h_1 x. \tag{8}$$

Suppose that each $x \in var(u)$ satisfies

$$\begin{tikzcd}
f'_1 h_1 x \arrow[r, "R", "*"'] & t_{1,x} \arrow[dr, Rightarrow, "E-R", "\Delta"'] & \\
& & t_{2,x} \arrow[dl, "*"', "R"] \\
f'_2 h_2 x \arrow[r, "*", "R"'] & t_{3,x} &
\end{tikzcd} \tag{9}$$

for some $t_{i,x} \in G(A)$, $i = 1,2,3$. Then we obtain (2):

$$\begin{tikzcd}
t_1 = f'_1 u_1 = f'_1 h_1 u \arrow[r, "R", "*"'] & {} \arrow[dr, Rightarrow, "E-R", "\Delta"'] & \\
& & {} \arrow[dl, "*"', "R"] \\
t_2 = f'_2 u_2 = f'_2 h_2 u \arrow[r, "*", "R"'] & {} &
\end{tikzcd}$$

Hence it remains to show (9).
Let $x \in var(u)$.

<u>Case 1</u>: $x \in V_1$. If $fx = f'_2 h_2 x$, then

$$f'_2 h_2 x = fx = f_1 x = f_1 x_1 = gl \xrightarrow[R]{} gr$$
$$= f'_1 x_1 = f'_1 h_1 x_1 = f'_1 h_1 x.$$

Choosing $t_{i,x} = f'_1 h_1 x$ for $i = 1,2,3$, we conclude (9) from

$$f'_1 h_1 x = f'_1 x \subseteq f'_1 u_1 = t_1 \in G(A).$$

Let $fx \neq f'_2 h_2 x$. Then by (7),

$$fx \xRightarrow[E-R]{} f'_2h_2x. \tag{10}$$

Since $fx = f_1x_1 \subseteq f_1u_1 = t\epsilon G(A)$, (10) and Prop. 10.13 imply

$$fx \xRightarrow[G(E-R)]{} f'_2h_2x.$$

We set

$$BR = \begin{cases} E-R & \text{if CRIT}(\{<l,r>\},E-R) \text{ is relatively} \\ & <R,A>\text{-unifiable and not including} \\ G(E-R) & \text{otherwise} \end{cases}$$

and obtain

$$fx \xRightarrow[BR]{} f'_2h_2x.$$

By Lemma 10.9, there are $h,h'\epsilon Z(T(A))$ and a linear term u' with $hu' = fx$ and $h'u' = f'_2h_2x$. We have two cases:

(i) For all $y\epsilon var(u')$ some $z\epsilon var(l)$ satisfies $occ(z,l) \leq occ(y,u')$.

(ii) Some $y\epsilon var(u')$ satisfies $occ(z,l) \nleq occ(y,u')$ for all $z\epsilon var(l)$.

Since $gl = f_1x_1 \subseteq f_1u_1 = t\epsilon G(A)$, we may assume that $g\epsilon Z(G(A))$.

<u>Case 1.1</u>: (i) holds. By Lemma 10.9, $gl = fx \xRightarrow[BR]{} f'_2h_2x$ implies

$$\begin{tikzcd} gr \arrow[dr, Rightarrow, "BR"] & \\ & {} \\ f'_2h_2x \arrow[ur, "\{<l,r>\}"'] & \end{tikzcd}$$

We set $t_{1,x} = gr$ and conclude (9) from

$$\begin{tikzcd} f'_1h_1x = f'_1x = f'_1x_1 = gr \arrow[dr, Rightarrow, "E-R", "\Delta"'] & \\ & {} \\ f'_2h_2x \arrow[ur, "R"'] & \end{tikzcd}$$

<u>Case 1.2</u>: (ii) holds. By Lemma 10.10, $gl = fx \xRightarrow[BR]{} f'_2h_2x$ implies $\gamma\varphi l = fx$, $\gamma' v' = f'_2h_2x$ and $\gamma \overset{\Delta}{\underset{BR}{\Longrightarrow}} \gamma'$ for some $\gamma, \gamma' \in Z(G(A))$ and a critical pair $\langle v,v'\rangle$ of $\langle R,BR\rangle$ with generator $\langle\langle l,r\rangle,\varphi\rangle$. By assumption (a), $\langle v,v'\rangle$ is relatively $\langle R,A\rangle$-unifiable. Hence there are terms v_1, v_2 such that

$$v \xrightarrow[R]{*} v_1 \overset{\Delta}{\underset{E-R}{\Longrightarrow}} v_2 \tag{11}$$

and $\langle v_2,v'\rangle$ is absolutely $\langle R,A\rangle$-unifiable.
Let $x \in var(\varphi l)$.
By Prop. 10.13,

$$\gamma \overset{\Delta}{\underset{E-R}{\Longrightarrow}} \gamma' . \tag{12}$$

Thus (11) and (12) yield

$$\gamma v \xrightarrow[R]{*} \gamma v_1 \overset{\Delta}{\underset{E-R}{\Longrightarrow}} \gamma' v_2 . \tag{13}$$

Since R is normalizing, some $g' \in Z(G(A))$ satisfies

$$\gamma' z \xrightarrow[R]{*} g'z \in NF(R)$$

for all $z \in X$.
$\gamma\varphi l = fx = gl$ and $var(r) \subseteq var(l)$ imply $\gamma v = \gamma\varphi r = gr$.
Since $\langle v_2,v'\rangle$ is absolutely $\langle R,A\rangle$-unifiable, (9) follows from (13):

$$
\begin{tikzcd}
f_1'h_1x = f_1'x = f_1'x_1 = gr = \gamma v \arrow[rrr, "R", "*"'] & & & \gamma' v_1 \arrow[d, Rightarrow, "E-R", "\Delta"'] \\
& & & \gamma' v_2 \arrow[d, "*", "R"'] \\
f_2'h_2x = \gamma' v' \arrow[r, "*", "R"'] & g'v' \arrow[r, "*", "R"'] & {} & g'v_2 \arrow[l, "*"', "R"]
\end{tikzcd}
$$

Hence (9) holds for all $x \in V_1$.

Case 2: $x \in V_2$. If $fx = f'_1 h_1 x$, then

$$f'_1 h_1 x = fx = f_2 x = g_x l_x \xRightarrow[E-R]{} g'_x r_x$$
$$= f'_2 x = f'_2 h_2 x.$$

Choosing $t_{1,x} = f'_1 h_1 x$ and $t_{2,x} = t_{3,x} = f'_2 h_2 x$, we conclude (9) from

$$f'_2 h_2 x = f'_2 x \leq f'_2 u_2 = t_2 \in G(A).$$

Let $fx \neq f'_1 h_1 x$. Then by (8),

$$fx \xrightarrow[R]{} f'_1 h_1 x.$$

Since $fx = f_2 x \leq f_2 u_2 = t \in G(A)$, Lemma 9.7 implies that there are $h, h' \in Z(G(A))$ and a linear term u such that $var(u) = \{y\}$ for some $y \in X$, $hu = fx$ and $h'u = f'_1 h_1 x$. We have two cases:

(iii) $w \leq occ(y,u')$ for some occurrence w of a variable in l_x.

(iv) $w \not\leq occ(y,u')$ for all occurrences w of variables in l_x.

Since $g_x l_x = f_2 x \leq f_2 u_2 = t \in G(A)$, we may assume that $g_x \in Z(G(A))$.

Case 2.1: (iii) holds. By Lemma 9.9, $g_x l_x = fx \xrightarrow[R]{} f'_1 h_1 x$ implies $f'_1 h_1 x \xrightarrow[R]{\Delta} g' l_x$ for some $g' \in Z(G(A))$ with $g_x z \xrightarrow[R]{\Delta} g'z$ for all $z \in var(l_x)$.
Let $z \in var(l_x)$. Since l_x is not a variable,

$$g_x z \subset g_x l_x = f_2 x \leq f_2 u_2 = t \in G(A)$$

and thus $g_x z <_R t$. By induction hypothesis,

$$\begin{tikzcd}[column sep=small]
 & g'z \\
g_x z \arrow[ur, "R", "\Delta"'] \arrow[dr, Rightarrow, "E-R"', "\Delta"] & \\
 & g_x'z
\end{tikzcd}
\quad \text{implies} \quad
\begin{tikzcd}
g'z \arrow[r, "R", "*"'] & g_1'z \arrow[dr, Rightarrow, "E-R", "\Delta"'] & \\
 & & g_2'z \arrow[dl, "*"', "R"] \\
g_x'z \arrow[r, "*", "R"'] & g_3'z &
\end{tikzcd}$$

for some $g'_1, g'_2, g'_3 \in Z(G(A))$. Therefore (9) is obtained as follows:

$$\begin{tikzcd}
f_1'h_1x \arrow[r, "R", "\Delta"'] & g'l_x \arrow[r, "R", "*"'] & g_1'l_x \arrow[dr, Rightarrow, "E-R", "\Delta"'] & \\
 & & & g_2'r_x \arrow[dl, "*"', "R"] \\
f_2'h_2x = f_2'x = g_x'r_x \arrow[rr, "*", "R"'] & & g_3'r_x &
\end{tikzcd}$$

<u>Case 2.2</u>: (iv) holds. By Lemma 9.10, $g_x l_x = fx \xrightarrow[R]{} f'_1 h_1 x$ implies $\gamma\varphi l_x = fx$ and $\gamma v' = f'_1 h_1 x$ for some $\gamma \in Z(G(A))$ and a simple critical pair $\langle v, v' \rangle$ of $\langle E-R, R \rangle$ with generator $\langle\langle l_x, r_x \rangle, \varphi \rangle$. Since R is normalizing, some $f' \in Z(G(A))$ satisfies

$$\gamma z \xrightarrow[R]{*} f'z \in NF(R)$$

for all $z \in X$. By assumption (b), there is $g' \in Z(G(A))$ such that

$$f'v' \xrightarrow[R]{*} g'l_x$$

and for all $z \in var(l_x)$

$$f'\varphi z \xrightarrow[R]{*} g'z.$$

Let $z \in var(l_x)$. Since l_x is not a variable,

$$g_x z \subset g_x l_x = f_2 x \leq f_2 u_2 = t \in G(A)$$

and thus $g_x z <_R t$. Moreover, $g_x l_x = fx = \gamma\varphi l_x$ implies $g_x z = \gamma\varphi z$ so that

$$\begin{tikzcd}
 & f'\varphi z \arrow[r, "R", "*"'] & g'z \\
g_x z \arrow[ur, "R", "*"'] \arrow[dr, Rightarrow, "\Delta", "E-R"'] & & \\
 & g'_x z &
\end{tikzcd}$$

By induction hypothesis,

$$\begin{tikzcd}
g'z \arrow[r, "R", "*"'] & g'_1 z \arrow[d, Rightarrow, "E-R", "\Delta"'] \\
 & g'_2 z \arrow[d, "*", "R"'] \\
g'_x z \arrow[r, "*", "R"'] & g'_3 z
\end{tikzcd}$$

for some $g'_1, g'_2, g'_3 \in Z(G(A))$, and (9) results from

$$\begin{tikzcd}
f'_1 h_1 x = \gamma v' \arrow[r, "R", "*"'] & f'v' \arrow[r, "R", "*"'] & g'l_x \arrow[r, "R", "*"'] & g'_1 l_x \arrow[d, Rightarrow, "E-R"] \\
 & & & g'_2 r_x \arrow[d, "*", "R"'] \\
 & f_2' h_2 x = f_2' x = g_x' r_x \arrow[rr, "*", "R"'] & & g_3' r_x
\end{tikzcd}$$

Hence (9) holds for all $x \in V_2$.
Since $var(u) = V_1 \cup V_2$, all $x \in var(u)$ satisfy (9). □

Thms. 9.15, 10.7 and 10.14 amount to the following consistency criterion:

10.15 <u>Consistency Theorem</u>

Let either H be a linear, normalizing and base-complete relation on T such that for all $\langle l,r\rangle \in H$ $op(l) \cap POP = \emptyset$, or let $K = \{\emptyset\}$ and $H = \emptyset$.
Let R be a linear, normalizing, base-complete and base-consistent term rewriting system on T such that R includes H and E-R is a term rewriting system contained in BE.
If SCRIT(R,R) is absolutely R-unifiable, CRIT(R,E-R) is relatively R-unifiable and not including and CRIT(E-R,R) is empty, then PAR is consistent w.r.t. $\langle BPAR,K\rangle$. □

10.16 <u>Definition</u>

Given relations R and R' on T, $\langle t,t'\rangle \in T^2$ (resp. $M \subseteq T^2$) is called <u>relatively R-convergent w.r.t. R'</u> if (for all $\langle t,t'\rangle \in M$) there is $\langle u,u'\rangle \in R'$ such that

$$t \xrightarrow[R]{*} \overset{\Delta}{\underset{E-R}{\Longrightarrow}} \xrightarrow[R]{*} u \quad \text{and} \quad t' \xrightarrow[R]{*} u' \ .$$

Clearly, if $R \subseteq T^2$, and $M \subseteq T^2$ is relatively R-convergent w.r.t. = (equality on T), then M is relatively R-unifiable. This fact provides a "practical" consistency criterion derived from Thm. 10.15 and generalizing Thm. 9.18:

10.17 Consistency Theorem

Let E-BE be base-total and included in $T_{OP-POP} \times T$. Let $R \subseteq ((T-BT) \times T) \cup BE$ be linear and directly decreasing such that $E-BE \subseteq R$ and E-R is a term rewriting system. If SCRIT(R,R) is absolutely R-convergent w.r.t. =, CRIT(R,E-R) is relatively R-convergent w.r.t. = and not including and if CRIT(E-R,R) is empty, then PAR is consistent w.r.t. $\langle BPAR,K \rangle$.

Proof:

Set H = E-BE. Since $E-BE \subseteq R$, H is linear and directly decreasing. Thus by Thms. 6.10, 6.5 and Coroll. 5.4(i), H and R are normalizing. By Thm. 7.2(i), H is base-complete. Since $R \subseteq ((T-BT) \times T) \cup BE$, Lemma 4.18 implies that R is base-consistent. Hence the conjecture follows from Thm. 10.15. □

11. Consistent specifications with conditionals

This chapter resumes the case we studied in chapter 8, namely that for all $A \in K$ BSPEC(A) is a correct extension of bool. Again, the set of base predicates (terms in BT_{bool}) is denoted by BP, and a set C of conditionals (cf. 8.2) is fixed. Furthermore, we recall Prop. 8.1, which says that BSPEC(A) is a correct extension of bool if and only if bool is a subspecification of BSPEC(A) such that for all $t \in BG(A)_{bool}$ either $t \equiv_{BSPEC(A)} TRUE$ or $t \equiv_{BSPEC(A)} FALSE$.

So assume that for all $A \in K$ BSPEC(A) is a correct extension of bool. We shall come up with two consistency criteria which differ from Thm. 9.18 resp. 10.17, essentially in that base-totality (w.r.t. BOP) and R-convergence w.r.t. equality on T are weakened to base-totality w.r.t. BOP-C (cf. 8.4) and R-convergence w.r.t. conditional equality (Def. 11.1), respectively. The crucial fact is that R-convergence w.r.t. conditional equality implies $<R \cup CR(A), A>$-unifiability where CR(A) is the set of conditional A-rules (Def. 11.4, Thm. 11.6).

We proceed with two criteria for absolute resp. relative confluence of $<R \cup CR(A) \cup CCR, A>$ (Thms. 11.8, 11.9) weakening the criteria for confluence of $<R, A>$ given by Thms. 9.15 and 10.14. Thms. 11.10 and 11.11 present consistency conditions derived from 11.8/9 and related to 9.18 and 10.17 as indicated above. Finally, we use Thm. 11.11 for a consistency proof of arrayl (cf. 2.2) and setl (cf. 8.7).

11.1 <u>Definition</u>

The sets <u>ct(t,p)</u>, $t \in T$, $p \in BP$, of <u>conditional subterms of t w.r.t. p</u> are inductively defined as follows:

$$ct(t,p) = \begin{cases} \{\langle t',q\wedge q'\rangle / \langle t',q\rangle \in ct(u,p)\} \\ \qquad \cup \{\langle t',q\wedge \neg q'\rangle / \langle t',q\rangle \in ct(u',p)\} \\ \qquad\quad \text{if } t=IF(q',u,u'),\ IF\in C \text{ and } q'\in BP \\ \{\langle t,p\rangle\} \text{ otherwise} \end{cases}$$

Note that in general neither $ct(t,p) \subseteq sct(t,p)$ nor $sct(t,p) \subseteq ct(t,p)$ (cf. Def. 8.9).

$p\in BP$ is <u>contradictory</u> if for all $A\in K$ and $f\in BZ(G(A))$

$fp \equiv_{BSPEC(A)} FALSE.$

$t,t'\in T$ are <u>conditionally equal</u> if for all $\langle u,p\rangle\in ct(t,TRUE)$ and $\langle u',p'\rangle\in ct(t',TRUE)$ $u = u'$ or $p\wedge p'$ is contradictory. The set of all pairs $\langle t,t'\rangle\in T^2$ where t and t' are conditionally equal is called <u>conditional equality</u> and denoted by $\underset{\sim}{\sim}$.

For verification purposes we define a family of binary relations $\underset{p}{\sim}$, $p\in BP$, on T by:

$t \underset{p}{\sim} t'$ iff for all $\langle u,q\rangle\in ct(t,p)$ and $\langle u',q'\rangle\in ct(t',p)$
$u = u'$ or $p\wedge q\wedge q'$ is contradictory.

Clearly, $\underset{TRUE}{\sim} = \underset{\sim}{\sim}$.

11.2 <u>Proposition</u>

$\{\underset{p}{\sim}\}_{p\in BP}$ satisfies the following recursive definition:

Let $IF\in C, q\in BP, u_1, u_2, u_2', t_1, t_2, IF(q,t_1,t_2)\in T$ such that for $i = 1,2$ $root(u_i)\in C$ implies $arg(u_i)_1 \in BP$.

(i) $IF(q,t_1,t_2) \underset{p}{\sim} u_2'$ iff $t_1 \underset{p\wedge q}{\sim} u_2'$ and

$t_2 \underset{p\wedge\neg q}{\sim} u_2$,

(ii) $u_1 \underset{p}{\sim} IF(q,t_1,t_2)$ iff $u_1 \underset{p\wedge q}{\sim} t_1$ and $u_1 \underset{p\wedge\neg q}{\sim} t_2$,

(iii) $u_1 \underset{p}{\sim} u_2$ iff $u_1 = u_2$ or p is contradictory.

Proof:

(i) Let $IF(q,t_1,t_2) \underset{p}{\sim} u'_2$. Then for all $\langle u,q'\rangle \epsilon ct(IF(g,t_1,t_2),p)$ and $\langle u',q''\rangle \epsilon ct(u'_2,p)$
$u = u'$ or $p\wedge q'\wedge q''$ ist contradictory.
We have to conclude $t_1 \underset{p\wedge q}{\sim} u'_2$ and $t_2 \underset{p\wedge\neg q}{\sim} u'_2$. So let $\langle u,q'\rangle \epsilon ct(t_1,p\wedge q)$, $\langle t,p'\rangle \epsilon ct(t_2,p\wedge\neg q)$,

$$\langle u',q''\rangle \epsilon ct(u'_2,p\wedge q) \text{ and }$$
$$\langle t',p''\rangle \epsilon ct(u'_2,p\wedge\neg q). \qquad (1)$$

Hence $\langle u,q'\wedge q\rangle \epsilon ct(IF(q,t_1,t_2),p\wedge q)$ and $\langle t,p'\wedge\neg q\rangle \epsilon ct(IF(q,t_1,t_2),p\wedge\neg q)$ so that

$$\langle u,q'\rangle, \langle t,p'\rangle \epsilon ct(IF(q,t_1,t_2),p). \qquad (2)$$

Analogously, (1) implies

$$\langle u',q''\rangle, \langle t',p''\rangle \epsilon ct(IF(q,u'_2,u'_2),p).$$

Thus there are q''_o, $p''_o \epsilon BP$ such that

$$\langle u',q''_o\rangle, \langle t',p''_o\rangle \epsilon ct(u'_2,p) \qquad (3)$$

and w.l.o.g. $q'' = q''_o\wedge q$ and $p'' = p''_o\wedge q$. By assumption, (2) and (3) imply

$$u = u' \text{ or } p\wedge q'\wedge q''_o \text{ is contradictory}$$

and

$t = t'$ or $p \wedge p' \wedge p''_o$ is contradictory.

Hence

$u = u'$ or $p \wedge q \wedge q' \wedge q''$ is contradictory

and

$t = t'$ or $p \wedge \neg q \wedge p' \wedge p''$ is contradictory.

Therefore, $t_1 \underset{p \wedge q}{\sim} u'_2$ and $t_2 \underset{p \wedge \neg q}{\sim} u'_2$.
Vice versa, let $t_1 \underset{p \wedge q}{\sim} u'_2$, $t_2 \underset{p \wedge \neg q}{\sim} u'_2$, $\langle u, q' \rangle \in ct(IF(q, t_1, t_2), p)$ and $\langle u', q'' \rangle \in ct(u'_2, p)$.
Then there is $q'_o \in BP$ such that

$\langle u, q'_o \rangle \in ct(t_1, p)$ and $q' = q'_o \wedge q$

or

$\langle u, q'_o \rangle \in ct(t_2, p)$ and $q' = q'_o \wedge \neg q$.

By assumption, both cases imply

$u = u'$ or $p \wedge q'_o \wedge q''$ is contradictory.

Hence

$u = u'$ or $p \wedge q' \wedge q''$ is contradictory.

Therefore, $IF(q, t_1, t_2) \underset{p}{\sim} u'_2$.

(ii) Since for all $p \in BP$ $\underset{p}{\sim}$ is symmetric, (ii) follows from (i).

(iii) Since for $i = 1,2$ $ct(u_i,p) = \{\langle u_i,p\rangle\}$, (iii) is an immediate consequence of the definition of $\underset{p}{\sim}$. □

11.3 Proposition

(a) If C is empty, then $\sim$ is the equality on T.

(b) Let $p,p' \in BP$ and for all $A \in K$ and $f \in BZ(G(A))$ $fp \equiv_{BSPEC(A)} fp'$. Then $\underset{p}{\sim} = \underset{p'}{\sim}$.

(c) For all $p \in BP$ $\underset{p}{\sim}$ is reflexive.

(d) If $p \in BP$ is contradictory, then $\underset{p}{\sim} = T^2$.

Proof:

(a) Let $C = \emptyset$. Then for all $t \in T$ $ct(t,TRUE) = \{\langle t,TRUE\rangle\}$. Hence t t' implies $t = t'$ because TRUE is not contradictory. The converse follows from (c).

(b) is proved by induction on the recursive definition of $\underset{p}{\sim}$ given by Prop. 11.2:

(i) Since by assumption $f(p\wedge q) \equiv_{BSPEC(A)} f(p'\wedge q)$ and $f(p\wedge\neg q) \equiv_{BSPEC(A)} f(p'\wedge\neg q)$, we get by induction hypothesis

$$\begin{aligned} & IF(q,t_1,t_2) \underset{p}{\sim} u'_2 \\ \text{iff} \quad & t_1 \underset{p\wedge q}{\sim} u'_2 \text{ and } t_2 \underset{p\wedge\neg q}{\sim} u'_2 \\ \text{iff} \quad & t_1 \underset{p'\wedge q}{\sim} u'_2 \text{ and } t_2 \underset{p'\wedge\neg q}{\sim} u'_2 \\ \text{iff} \quad & IF(q,t_1,t_2) \underset{p'}{\sim} u'_2. \end{aligned}$$

(ii) Analogously to (i).

(iii) $u_1 \underset{p}{\sim} u_2$

iff $u_1 = u_2$ or p is contradictory

iff $u_1 = u_2$ or p' is contradictory

iff $u_1 \underset{p'}{\sim} u_2$.

(c) $t \underset{p}{\sim} t$ for all $t \in T$ is proved by induction on size(t): If root(t)$\in$C implies $\arg(t)_i \notin BP$, then $t \underset{p}{\sim} t$ follows from 11.2(iii). Otherwise $t = IF(q,t_1,t_2)$ for some IF$\in$C, q$\in$BP and $t_1,t_2 \in T$. By induction hypothesis and 11.2(iii), $t_1 \underset{p \wedge q}{\sim} t_1$ and $t_2 \underset{FALSE}{\sim} t_1$. Hence (b) yields $t_1 \underset{p \wedge q \wedge q}{\sim} t_1$ and $t_2 \underset{p \wedge q \wedge \neg q}{\sim} t_1$. Thus $t \underset{p \wedge q}{\sim} t_1$ by 11.2(i). Analogously, $t \underset{p \wedge \neg q}{\sim} t_2$. Again by 11.2(i), $t \underset{p}{\sim} t$.

(d) immediately follows from the fact that if p is contradictory, then for all q BP p q is contradictory, too. □

11.4 Definition

Let A$\in$K. The term relation CR(A) of conditional A-rules consists of all linear rules

$$<IF(p,x,y),x> \text{ and } <IF(q,x,y),y>$$

with IF$\in$C, $p \equiv_{BSPEC(A)} TRUE$, $q \equiv_{BSPEC(A)} FALSE$ and $x,y \in X$.

11.5 Lemma

Let R be a base-complete relation on T and $t \sim t'$. Then for all A$\in$K <t,t'> is absolutely

$\langle CR(A), A \rangle$ - unifiable (cf. 9.12).

<u>Proof:</u>

For all $p \in BP$ a binary relation $\underset{p}{\approx}$ on T is defined as follows:

$t \underset{p}{\approx} t'$ iff for all $A \in K$ and $f \in Z(G(A))$ with $fX \subseteq NF(R)$

$$fp \equiv_{BSPEC(A)} TRUE \text{ implies } \begin{tikzcd}[row sep=small] ft \arrow[dr, "CR(A)", "*"'] & \\ & {} \\ ft' \arrow[ur, "CR(A)"', "*"] & \end{tikzcd}$$

We show $\underset{p}{\sim} \subseteq \underset{p}{\approx}$ by induction on the recursive definition of $\underset{p}{\sim}$ (cf. 11.2) and thus obtain the conjecture of this lemma because $t \underset{TRUE}{\approx} t'$ means that for all $A \in K$ $\langle t,t' \rangle$ is absolutely $\langle CR(A), A \rangle$ - unifiable.
So we enter into Prop. 11.2 (i) - (iii) and proceed as follows:

(i) $IF(q,t_1,t_2) \underset{p}{\sim} u'_2$ implies $t_1 \underset{p \wedge q}{\sim} u'_2$ and $t_2 \underset{p \wedge \neg q}{\sim} u'_2$.
Hence by induction hypothesis, for all $A \in K$ and $f \in Z(G(A))$ with $fX \subseteq NF(R)$

$$f(p \wedge q) \equiv_{BSPEC(A)} TRUE \text{ implies } \begin{tikzcd}[row sep=small] ft_1 \arrow[dr, "CR(A)", "*"'] & \\ & {} \\ fu'_2 \arrow[ur, "CR(A)"', "*"] & \end{tikzcd} \qquad (1)$$

and

$$f(p \wedge \neg q) \equiv_{BSPEC(A)} \text{TRUE implies} \quad \begin{array}{l} ft_2 \xrightarrow[*]{CR(A)} \cdot \\ fu'_2 \xrightarrow[*]{CR(A)} \cdot \end{array} \qquad (2)$$

Let $A \in K$ and $f \in Z(G(A))$ such that $fX \subseteq NF(R)$ and $fp \equiv_{BSPEC(A)} \text{TRUE}$. Base-completeness of R implies that $fq \in BG(A)$. If $fq \equiv_{BSPEC(A)} \text{TRUE}$, then $f(p \wedge q) \equiv_{BSPEC(A)} \text{TRUE}$. Hence by (1),

$$f(IF(q,t_1,t_2)) = IF(fq,ft_1,ft_2) \xrightarrow{CR(A)} ft_1 \xrightarrow[*]{CR(A)} \cdot \xleftarrow[*]{CR(A)} fu'_2$$

Otherwise $fq \equiv_{BSPEC(A)} \text{FALSE}$ which yields $f(p \wedge \neg q) \equiv_{BSPEC(A)} \text{TRUE}$ and thus by (2),

$$f(IF(q,t_1,t_2)) = IF(fq,ft_1,ft_2) \xrightarrow{CR(A)} ft_2 \xrightarrow[*]{CR(A)} \cdot \xleftarrow[*]{CR(A)} fu'_2$$

Therefore, $IF(q,t_1,t_2) \underset{p}{\approx} u'_2$.

(ii) By symmetry of $\underset{p}{\sim}$ and $\underset{p}{\approx}$, $u_1 \underset{p}{\sim} IF(q,t_1,t_2)$ implies $u_1 \underset{p}{\approx} IF(q,t_1,t_2)$.

(iii) If $u_1 \underset{p}{\sim} u_2$, then $u_1 = u_2$ or p is contradictory. In the first case $u_1 \underset{p}{\approx} u_2$ is clear. In the second case base-completeness of R implies $fp \equiv_{BSPEC(A)} \text{FALSE}$ for all $A \in K$ and $f \in Z(G(A))$ with $fX \subseteq NF(R)$. Thus

$u_1 \underset{p}{\approx} u_2$ is trivially true. □

11.6 Theorem

Let R be a base-complete relation on T and $M \subseteq T^2$ be absolutely (resp. relatively) R-convergent w.r.t. $\sim$ (cf. 9.17, 10.16). Then for all $A \in K$ M is absolutely (resp. relatively) $\langle R \cup CR(A), A \rangle$-unifiable (cf. 9.12, 10.11).

Proof:

Let $\langle t,t' \rangle \in M$, $A \in K$ and $f \in Z(G(A))$ with $fX \subseteq NF(R)$. Then there are two conditionally equal terms u,u' such that $t \xrightarrow[R]{*} u$ resp. $t \xrightarrow[R]{*} \overset{\Delta}{\underset{E-R}{\Longrightarrow}} u_o \xrightarrow[R]{*} u$ for some $u_o \in T$, and $t' \xrightarrow[R]{*} u'$. By Lemma 11.5,

$$\begin{tikzcd} fu \arrow[dr, "R\cup CR(A)", "*"'] & \\ & {} \\ fu' \arrow[ur, "*", "R\cup CR(A)"'] & \end{tikzcd}$$

Therefore,

$$\begin{tikzcd} ft \arrow[dr, "R\cup CR(A)", "*"'] & & & fu_o \arrow[dr, "R\cup CR(A)", "*"'] & \\ & {} & \text{resp.} & & {} \\ ft' \arrow[ur, "*", "R\cup CR(A)"'] & & & ft' \arrow[ur, "*", "R\cup CR(A)"'] & \end{tikzcd}$$

Hence $\langle t,t' \rangle$ is absolutely (resp. relatively) $\langle R \cup CR(A), A \rangle$-unifiable. □

R-convergence w.r.t. conditional equality is the main assumption in Consistency Thms. 11.10 and 11.11. Thus the following decidability criterion for R-convergence w.r.t. $\sim$ is important. □

11.7 <u>Proposition</u>

Let R be a normalizing relation on T and CONTRA be the set of all contradictory base predicates. Then absolute and relative R-convergence w.r.t. $\sim$ is decidable whenever CONTRA is decidable.

<u>Proof:</u>

Since R is normalizing, König's Infinity Lemma (cf. Knuth /42/, p. 381 - 383) implies that for all $t \in T$ the set $succ(t) = \{t' \in T / t \xrightarrow[R]{*} t'\}$ is finite. Hence

$$succ_o(t) = \{t' \in T / t \xrightarrow[R]{*} \overset{\Delta}{\underset{E-R}{\Longrightarrow}} \xrightarrow[R]{*} t'\}$$

is finite, too. Thus absolute resp. relative R-convergence w.r.t. $\sim$ of some $\langle t,t'\rangle \in T^2$ can be decided as follows provided that CONTRA is decidable:

Compute $M = succ(t) \times succ(t')$ resp. $M = succ_o(t) \times succ(t')$ and use the recursive definition of $\sim_p$ (11.2) in order to identify the intersection N of M and $\sim$. $\langle t,t'\rangle$ is absolutely (relatively) R-convergent w.r.t. $\sim$ iff N is not empty. □

11.8 <u>Absolute-confluence Theorem</u>

Let $R \subseteq T_{OP-C} \times T$ be a base-consistent term rewriting system. For all $A \in K$ $\langle R \cup CR(A) \cup CCR, A\rangle$ is absolutely confluent (cf. 8.3, 4.15) if

(i) $R \cup CCR$ is base-complete and contained in some simplification ordering (cf. 5.2),

(ii) $SCRIT(R,R)$ is absolutely $(R \cup CCR)$-convergent w.r.t. conditional equality.

<u>Proof:</u>

Let $A \in K$, $R(A) = R \cup CR(A) \cup CCR$ and R' be a simplification ordering that includes $R \cup CCR$. Since for all $\langle l,r \rangle \in CR(A)$ $l \supset r$, $CR(A)$ is contained in R'_A (cf. Def. 4.5). Hence $R(A) \subseteq R'_A$ so that by Corollary 5.4 (ii), $R(A)$ is normalizing. By Thm. 9.15, it remains to show that $SCRIT(R(A),R(A))$ is $\langle R(A),A \rangle$--unifiable.
So let $\langle t,t' \rangle \in SCRIT(R(A),R(A))$. By Def. 9.3 there are $\langle l,r \rangle \in R(A)$, $u \in T(A)$ and $g \in Z(T(A))$ with $gu = l$. Moreover, the set $\{z \in var(u) / gz \notin X\}$ consists of one element, say $\bar{x}$, and there is $\langle l',r' \rangle \in R(A)_X$ such that $\langle g\bar{x}, l' \rangle$ has a most general unifier $\langle \varphi, \beta \rangle$. Since $CRIT(CCR,R) \subseteq SCRIT(R,CCR)^{-1}$, we have one of the following six cases:

1. $\langle l,r \rangle \in R$, $\langle l',r' \rangle \in R_X$
2. $\langle l,r \rangle \in R$, $\langle l',r' \rangle \in CR(A)$
3. $\langle l,r \rangle \in R$, $\langle l',r' \rangle \in CCR$
4. $\langle l,r \rangle \in CR(A)$, $\langle l',r' \rangle \in R(A)_X$
5. $\langle l,r \rangle \in CCR$, $\langle l',r' \rangle \in CR(A)$
6. $\langle l,r \rangle, \langle l',r' \rangle \in CCR$.

In <u>case 1</u>, the absolute $\langle R(A),A \rangle$-unifiability of $\langle t,t' \rangle$ follows from Thm. 11.6 because $R \cup CCR$ is base-complete and $\langle t,t' \rangle$ is absolutely $(R \cup CCR)$--convergent w.r.t. $\sim$.

In <u>case 2</u>, the root of l' would be some conditional IF. Since $\varphi\beta g\bar{x} = \varphi l'$ and $g\bar{x} \in X$, we would get

$$IF = root(g\bar{x}) \in op(gu) = op(l)$$

in contradiction to $R \subseteq T_{OP-C} \times T$. Hence (b) cannot occur.

In <u>case 3</u>,

$$l' = \sigma(v, IF(b,x,y), w)$$

and

$$r' = IF'(b, \sigma(v,x,w), \sigma(v,y,w))$$

for some $\sigma \in OP-C$, $IF, IF' \in C$, $b,x,y \in X$ and $v,w \in X^*$. Since $\varphi\beta g\bar{x} = \varphi l'$, $g\bar{x} \notin X$ and $IF \notin op(l) = op(gu)$, there are $t_1, t_2 \in T^*$ and $a \in X$ with $g\bar{x} = \sigma(t_1, a, t_2)$. Thus $\langle\varphi, \beta\rangle$ is a most general unifier of $\langle a, IF(b,x,y)\rangle$. Since $\beta a \notin \{b,x,y\}$, there is another unifier of $\langle a, IF(b,x,y)\rangle$, namely $\langle\psi, \beta\rangle$ with $\psi\beta a = IF(b,x,y)$ and $\psi z = z$ for all $z \in X - \{\beta a\}$. But $\psi b = b$ implies that φb is a variable, otherwise $\langle\varphi, \beta\rangle$ would not be a <u>most general</u> unifier of $\langle a, IF(b,x,y)\rangle$.

Now let $f \in Z(G(A))$ with $fX \subseteq NF(R(A))$. Since $R(A)$ is base-complete, we have $f\varphi b \in BG(A)_{\underline{bool}}$ and thus w.l.o.g. $f\varphi b \equiv_{BSPEC(A)} TRUE$. Therefore,

$$f\varphi\beta a = f\varphi IF(b,x,y) = IF(f\varphi b, f\varphi x, f\varphi y) \xrightarrow[CR(A)]{} f\varphi x. \qquad (1)$$

Furthermore,

$$f\varphi r' = IF'(f\varphi b, f\varphi\sigma(v,x,w), f\varphi\sigma(v,y,w)) \xrightarrow[CR(A)]{} f\varphi\sigma(v,x,w).$$

Hence

$$f\varphi g'\bar{x} = f\varphi r' \xrightarrow[CR(A)]{} f\varphi\sigma(v,x,w) = f\varphi g''\bar{x} \qquad (2)$$

for $g',g''\in Z(T)$ with $g'\bar{x} = r'$, $g''\bar{x} = \sigma(v,x,w)$ and $g'z = g''z = \beta gz$ for all $z\in X-\{\bar{x}\}$. Since $t' = g'u$ (cf. 9.3), (2) implies

$$ft' = f\varphi g'u \xrightarrow[CR(A)]{} f\varphi g''u. \tag{3}$$

(1) yields $f\varphi\beta \xrightarrow[CR(A)]{} h$ for $h\in Z(T)$ with $ha = f\varphi x$ and $hz = f\varphi\beta z$ for all $z\in X-\{a\}$. Thus

$$ft = f\varphi\beta r \xrightarrow[CR(A)]{\Delta} hr. \tag{4}$$

Moreover,
$\varphi\beta g\bar{x} = \varphi l'$ implies $\varphi\beta t_1 = \varphi v$, $\varphi\beta t_2 = \varphi w$
and thus

$$\begin{aligned} &f\varphi g''\bar{x} = f\varphi\sigma(v,x,w) = \sigma(f\varphi\beta t_1, f\varphi x, f\varphi\beta t_2)\\ &= \sigma(f\varphi\beta t_1, ha, f\varphi\beta t_2) \xrightarrow[CR(A)]{\Delta} \sigma(ht_1, ha, ht_2)\\ &= hg\bar{x}. \end{aligned}$$

Hence

$$f\varphi g''u \xrightarrow[CR(A)]{\Delta} hgu = hl \xrightarrow[R]{} hr \tag{5}$$

because for all $z\in X-\{\bar{x}\}$

$$f\varphi g''z = f\varphi\beta gz \xrightarrow[CR(A)]{\Delta} hgz.$$

(4), (3) and (5) imply

$$\begin{tikzcd} ft \arrow[dr, "R'", "\Delta"'] & \\ & hr. \\ ft' \arrow[ur, "*", "R'"'] & \end{tikzcd}$$

Therefore, <t,t'> is absolutely <R(A),A>-unifiable.

Case 4 yields w.l.o.g.

$$l = IF(p,x,y) \quad \text{and} \quad r = x$$

where $IF\in C$, $p \equiv_{BSPEC(A)} TRUE$ and $x,y\in X$. We have two subcases:

Case 4.1: $u = \bar{x}$. Then

$$\varphi l' = \varphi\beta g\bar{x} = \varphi\beta gu = \varphi\beta l = IF(p,\varphi\beta x,\varphi\beta y).$$

Hence $root(l')\in C$ and thus $\langle l',r'\rangle\in CR(A)$. Therefore

$$l' = IF(p,x',y'), \qquad r' = x'$$

and $\varphi\beta x = \varphi x'$ for some $x',y'\in X$. Define $h\in Z(T(A))$ by

$$hz = \begin{cases} r' & \text{if } z = \bar{x} \\ \beta gz & \text{otherwise.} \end{cases}$$

Then

$$t = \varphi\beta r = \varphi\beta x = \varphi x' = \varphi r' = \varphi h\bar{x} = \varphi hu = t'$$

so that trivially, $\langle t,t'\rangle$ is absolutely $\langle R(A),A\rangle$--unifiable.

Case 4.2: $u \neq \bar{x}$. Since $\bar{x}\in var(u)$, u is not a variable. Thus $gu = l = IF(p,x,y)$ implies $u = IF(q,x',y')$ for some $q\in BT(A)$ and $x',y'\in X$. Since $gx' = x$ and $gy' = y$, but $g\bar{x}\notin X$, we have $\bar{x}\in var(q)$. Hence $g\bar{x}\leq gq = p$ implies $g\bar{x}\in BG(A)$. Therefore $gx = \varphi\beta gx = \varphi l'$, and we obtain

$$p = gq \xrightarrow[R']{\Delta} g'q \tag{6}$$

for $g' \in Z(T(A))$ with $g'\bar{x} = \varphi r'$ and $g'z = gz$ for all $z \in X-\{\bar{x}\}$. Define $h \in Z(T(A))$ as in case 4.1. Then

$$\varphi h\bar{x} = \varphi r' = g'\bar{x}. \tag{7}$$

Since $q \subset u$ and for all $z \in X-\{\bar{x}\}$ $gz \in X$, but $var(gq) = \emptyset$, $\bar{x}$ is the only variable of q so that by (7),

$$\varphi hq = g'q. \tag{8}$$

Since R is base-consistent, Lemma 4.18 and Thm. 8.5 (a) imply that $\langle R(A),A\rangle$ is base-consistent, too. Hence by (6),

$$g'q \equiv_{BSPEC(A)} p \equiv_{BSPEC(A)} TRUE. \tag{9}$$

We have

$$hx' = \beta gx' = \beta x \tag{10}$$

because $gx' = x$ and $g\bar{x} \notin X$ imply $x' \neq \bar{x}$. Thus by (8), (9) and (10),

$$t' = \varphi hu = \sigma(\varphi hq, \varphi hx', \varphi hy') = \sigma(g'q, \varphi hx', \varphi hy')$$
$$\xrightarrow[CR(A)]{} \varphi hx' = \varphi \beta x = \varphi \beta r = t.$$

Therefore, $\langle t,t'\rangle$ is absolutely $\langle R(A),A\rangle$-unifiable.

<u>Case 5</u> yields w.l.o.g.

$$\varphi \beta l = \sigma(v, IF(p,x,y), w),$$
$$t = IF'(p, \sigma(v,x,w), \sigma(v,y,w))$$

and

$$t' = \sigma(v,x,w)$$

where $\sigma \in OP\text{-}C$, $IF, IF' \in C$, $p \equiv_{BSPEC(A)} TRUE$, $x,y \in X$ and $v,w \in X^*$. Hence $t \xrightarrow[CR(A)]{} t'$, and $\langle t,t'\rangle$ is absolutely $\langle R(A),A\rangle$-unifiable.

Case 6 implies $u = \bar{x}$ and either $t = t'$ or

$$\varphi\beta 1 = \sigma(v, IF_1(b_1,x_1,y_1),w_1,IF_2(b_2,x_2,y_2),w_2),$$

$$t = IF_1'(b_1,\sigma(v,x_1,w_1,IF_2(b_2,x_2,y_2,),w_2), \sigma(v,y_1,w_1,IF_2(b_2,x_2,y_2),w_2))$$

and

$$t' = IF_2'(b_2,\sigma(v,IF_1(b_1,x_1,y_1),w_1,x_2,w_2), \sigma(v,IF_1(b_1,x_1,y_1),w_1,y_2,w_2))$$

for some $\sigma\epsilon OP\text{-}C$, $IF_i, IF_i'\epsilon C$, $b_i,x_i,y_i\epsilon X$ and $v,w_i\epsilon X^*$ where $i = 1,2$. Therefore

$$t \xrightarrow[CCR]{*} t_1 = IF_1'(b_1, IF_2'(b_2,\sigma(v,x_1,w_1,x_2,w_2), \sigma(v,x_1,w_1,y_2,w_2)), IF_2'(b_2,\sigma(v,y_1,w_1,x_2,w_2), \sigma(v,y_1,w_1,y_2,w_2)))$$

and

$$t \xrightarrow[CCR]{*} t_2 = IF_2'(b_2, IF_1'(b_1,\sigma(v,x_1,w_1,x_2,w_2), \sigma(v,y_1,w_1,x_2,w_2)), IF_1'(b_1,\sigma(v,x_1,w_1,y_2,w_2), \sigma(v,y_1,w_1,y_2,w_2))).$$

By Lemma 11.13 below, t_1 and t_2 are conditionally equal. Hence $<t,t'>$ is absolutely CCR-convergent w.r.t. $\sim$ and thus absolutely $<R(A),A>$-unifiable by Thm. 11.6. □

11.9 Relative-confluence Theorem

Let R be a linear term rewriting system on T and $R' = R\cup CR\cup CCR$ (cf. 8.2). For all $A\epsilon K$ $<R\cup CR(A)\cup CCR,A>$ is relatively confluent (cf. lo.6) if

(i) $R \cup CCR$ is base-complete and contained in some simplification ordering,

(ii) $E-R'$ is a term rewriting system and contained in $BE \cap (T_{OP-C} \times T)$,

(iii) $CRIT(R, E-R')$ is relatively $(R \cup CCR)$-convergent w.r.t. conditional equality and not including, $CRIT(E-R', R)$ and $CRIT(CCR, E-R')$ are empty,

(iv) $\langle R \cup CR(A) \cup CCR, A \rangle$ is absolutely confluent.

Proof:

Let $A \in K$, $R(A) = R \cup CR(A) \cup CCR$ and R'' be a simplification ordering that includes $R \cup CCR$. Since for all $\langle l, r \rangle \in CR(A)$ $l \supset r$, $CR(A)$ is contained in R''_A (cf. 4.5). Hence $R(A) \subseteq R''_A$ so that by Corollary 5.4 (ii), $R(A)$ is normalizing. By Thm. 10.14, the following three conditions are sufficient for the relative confluence of $\langle R(A), A \rangle$ because $E-R(A) \subseteq E-R'$:

(a) $CRIT(R \cup CCR, E-R')$ is relatively $\langle R(A), A \rangle$-unifiable and not including.

(b) $CRIT(CR(A), G(E-R'))$ is relatively $\langle R(A), A \rangle$-unifiable (cf. Def. lo.12).

(c) For all $\langle t, t' \rangle \in SCRIT(E-R', R(A))$ and $f \in Z(G(A))$ with $fX \subseteq NF(R(A))$ some $h \in Z(G(A))$ satisfies

$$ft' \xrightarrow[R(A)]{*} hl \text{ and } f\varphi \xrightarrow[R(A)]{*} h$$

where $\langle \langle l, r \rangle, \varphi \rangle$ is a generator of $\langle t, t' \rangle$.

Since $R \cup CCR$ is base-complete and $CRIT(R \cup CCR, E-R')$ is relatively $(R \cup CCR)$-convergent w.r.t. $\sim$, (a) follows from Thm. 11.6.

As to (b): Let $\langle t,t'\rangle \in CRIT(CR(A),G(E-R'))$. Then there are $\langle l,r\rangle \in CR(A)$, $u \in T(A)$ and $g \in Z(T(A))$ with $gu = l$. Moreover, the set $\{z \in var(u) / gz \notin X\} = \{x_1,\dots,x_n\}$ is nonempty, and there are $\langle l_1,r_1\rangle,\dots,\langle l_n,r_n\rangle \in G(E-R')$ such that $\langle\langle gx_1,\dots,gx_n\rangle,\langle l_1,\dots,l_n\rangle\rangle$ has a most general unifier $\langle \varphi,\beta\rangle$. Hence w.l.o.g.

$$l = IF(p,x,y) \text{ and } r = x$$

where $IF \in C$, $p \equiv_{BSPEC(A)} TRUE$ and $x,y \in X$.
If $u = x_i$ for some $1 \leq i \leq n$, then

$$\varphi l_i = \varphi\beta g x_i = \varphi\beta g u = \varphi\beta l = IF(p,\varphi\beta x,\varphi\beta y).$$

Thus $root(l_i) \in C$ in contradiction to (ii). Therefore $u \notin \{x_1,\dots,x_n\}$ so that $u \notin X$ because $\{x_1,\dots,x_n\} \subseteq var(u)$. Hence $gu = l = IF(p,x,y)$ implies $u = IF(y,x',y')$ for some $q \in BT(A)$ and $x',y' \in X$. Since $gx' = x$ and $gy' = y$, but $gx_i \notin X$ for all $1 \leq i \leq n$, we have $\{x_1,\dots,x_n\} \subseteq var(q)$. Thus $gx_i \subseteq gq = p \in BG(A)$ implies $gx_i = \varphi\beta g x_i = \varphi l_i$, and we obtain

$$p = gq \xrightarrow[G(E-R')]{} g'q \tag{1}$$

for $g' \in Z(T(A))$ with $g'x_i = \varphi r_i$ for all $1 \leq i \leq n$ and $g'z = gz$ for all $z \in X - \{x_1,\dots,x_n\}$. Define $h \in Z(T(A))$ by

$$hz = \begin{cases} r_i & \text{if } z = x_i \text{ for some } 1 \leq i \leq n \\ \beta g z & \text{otherwise.} \end{cases}$$

Since $q \subset u$ and for all $z \in X - \{x_1,\dots,x_n\}$ $gz \in X$, but $var(gq) = \emptyset$, we get $var(q) \subseteq \{x_1,\dots,x_n\}$, and thus $hx_i = \varphi r_i = g'x_i$, $1 \leq i \leq n$, implies

$$\varphi hq = g'q \tag{2}$$

By assumption (ii), (1) yields

$$g'q \equiv_{BSPEC(A)} p \equiv_{BSPEC(A)} TRUE.$$

Since $gx' = x$, but $gx_i \notin X$ for all $1 \leq i \leq n$, we have $x' \notin \{x_1, \ldots, x_n\}$ so that

$$hx' = \beta gx' = \beta x. \qquad (3)$$

Thus (1) - (3) imply

$$t' = \varphi hu = \sigma(\varphi hq, \varphi hx', \varphi hy') = \sigma(g'q, \varphi hx', \varphi hy')$$
$$\xrightarrow[CR(A)]{} \varphi hx' = \varphi\beta x = \varphi\beta r = t.$$

Therefore, $\langle t,t'\rangle$ is relatively $\langle R(A),A\rangle$ -unifiable.

<u>As to (c)</u>: Let $\langle t,t'\rangle \in SCRIT(E\text{-}R',R(A))$ with generator $\langle\langle l,r\rangle,\varphi'\rangle$. Since $CRIT(E\text{-}R',R)$ is empty and $E\text{-}R' \subseteq T_{OP-C} \times T$, $\langle t,t'\rangle$ must be a simple critical pair of $\langle E\text{-}R', CCR\rangle$. Thus we have $\langle l,r\rangle \in E\text{-}R'$, $u \in T(A)$, $g \in Z(T(A))$ and $\langle l',r'\rangle \in CCR$ such that $gu = l$, the set $\{z \in var(u) / gz \notin X\}$ consists of one element, say $\bar{x}$, and $\langle g\bar{x}, l'\rangle$ has a most general unifier $\langle \varphi, \beta\rangle$. We proceed with Thm. 11.8, case 3, and obtain for all $f \in Z(G(A))$ with $fX \subseteq NF(R(A))$

$$ft' \xrightarrow[CR(A)]{*} hl$$

(cf. 11.8 (3), (5)) and $f\varphi' = f\varphi\beta \xrightarrow[CR(A)]{} h$. □

11.10 <u>Consistency Theorem</u>

Let $R \subseteq ((T_{OP-C}\text{-}BT) \times T) \cup (BE \cap (T_{BOP-C} \times BT))$ be a directly decreasing term rewriting system. PAR is consistent w.r.t. $\langle BPAR,K\rangle$ if

(i) $E \subseteq R \cup CR \cup CCR$ (cf. 8.2/3),

(ii) E-BE is linear, base-total w.r.t. BOP-C (cf. 8.4) and contained in $T_{OP-POP} \times T$,

(iii) SCRIT(R,R) is absolutely (R∪CCR)-convergent w.r.t. conditional equality.

Proof:

We want to apply Thm. 4.19 in order to show the conjecture. So set H = (E-BE)∪CCR and R(A) = R∪CR(A)∪CCR for all A∈K. By (ii) and the definition of CCR (8.3), H is linear and for all <l,r>∈H op(l)∩POP = ∅. Since R is directly decreasing, Thm. 8.5(b) implies that R∪CCR is directly decreasing, too. Therefore, Thms. 6.5 and 6.10 provide a simplification ordering that includes R∪CCR and thus H because E-BE⊆R∪CCR. Hence H is normalizing by Coroll. 5.4 (i). Again by Thm. 8.5(b), base-totality of E-BE w.r.t. BOP-C implies base-totality of H (w.r.t BOP), and we conclude from Thm. 7.2 (i) that H is base-complete.

By (i), R(A) contains E∪H.

Absolute confluence of <R(A),A> follows from Thm. 11.8: Since R⊆((T-BT)xT)∪BE, R is base-consistent by Lemma 4.18. Above we have shown that H is base-complete. Thus R∪CCR is base-complete because it includes H. We have also seen that R∪CCR is contained in some simplification ordering. Hence 11.8 (i) is satisfied. 11.8 (ii) agrees with condition (iii) of the present theorem.

Since R is base-consistent, Lemma 4.18 and Thm. 8.5(a) imply that <R(A),A> is base-consistent, too. □

11.11 Consistency Theorem

Let $R \subseteq ((T_{OP-C}-BT) \times T) \cup (BE \cap (T_{BOP-C} \times BT))$ be a linear and directly decreasing term rewriting system. PAR is consistent w.r.t. <BPAR,K> if

(i) E-BE is linear, base-total w.r.t. BOP-C and contained in $R' \cap (T_{OP-POP} \times T)$,

(ii) E-R' is a term rewriting system and contained in $BE \cap (T_{OP-C} \times T)$,

(iii) SCRIT(R,R) is absolutely (R∪CCR)-convergent w.r.t. conditional equality, CRIT(R,E-R') is relatively (R∪CCR)-convergent w.r.t. conditional equality and not including, CRIT(E-R',R) and CRIT(CCR,E-R') are empty

where R' = R∪CR∪CCR (cf. 8.2/3).

<u>Proof:</u>

We want to apply Thm. 10.7 in order to show the conjecture. So let H = (E-BE)∪CCR and R(A) = R∪CR(A)∪CCR for all $A \in K$. By (i) and the definition of CCR (8.3), H is linear and for all $\langle l,r \rangle \in H$ $op(l) \cap POP = \emptyset$. Since R is directly decreasing, Thm. 8.5(b) implies that R∪CCR is directly decreasing, too. Therefore, Thms. 6.5 and 6.10 provide a simplification ordering R" that includes R∪CCR and thus H because $E\text{-}BE \subseteq R \cup CCR$. Hence H is normalizing by Coroll. 5.4(i). Again by Thm. 8.5(b), base-totality of E-BE w.r.t.BOP-C implies base-totality of H (w.r.t BOP), and we conclude from Thm. 7.2(i) that H is base-complete.

Since for all $\langle l,r \rangle \in CR(A)$ $l \supset r$, CR(A) is contained in R''_A (cf. 4.5). Hence $R(A) \subseteq R''_A$ so that by Coroll. 4.5 (ii), R(A) is normalizing. $E\text{-}BE \subseteq R' \subseteq R(A)$ and assumption (ii) imply that $H \subseteq R(A)$ and E-R(A) is a term rewriting system contained in BE.
Absolute confluence of $\langle R(A),A \rangle$ follows from Thm. 11.8: Since $R \subseteq ((T\text{-}BT) \times T) \cup BE$, R is base-consistent by Lemma 4.18. Above we have shown that H is base-complete. Thus R∪CCR is base-complete because it includes H. We have also seen that R∪CCR is contained in some simplification ordering.

Hence 11.8(i) is satisfied. 11.8(ii) follows from condition (iii) of the present theorem.

By Thm. 11.9, <R(A),A> is relatively confluent: 11.9(i) agrees with 11.8(i), which was shown above. 11.9(ii) coincides with condition (ii) of the present theorem. 11.9(iii) follows from assumption (iii). 11.9(iv) has just been proved.
We already know that $R \cup CCR$ is base-complete. Hence <R(A),A> is base-complete because R(A) includes $R \cup CCR$. Since R is base-consistent, Lemma 4.18 and Thm. 8.5(a) imply that <R(A),A> is base-consistent, too. □

We close this chapter by applying Thm. 11.11 to consistency proofs of the specifications array1 (cf. 2.2) and set1 (cf. 8.7). In particular, we have to show the convergence of some critical pairs w.r.t. conditional equality. For that purpose the following three lemmata will be useful since they state certain term schemata all instances of which are conditionally equal.

11.12 Lemma

For all $IF \in C$, $p,q \in B$ and $t,t' \in T$ with sort(IF) = sort(t) = sort(t'),

(i) $IF(q,t,t) \underset{p}{\sim} t$,

(ii) $IF(q,t,t') \underset{p \wedge q}{\sim} t$ and $IF(q,t,t') \underset{p \wedge \neg q}{\sim} t'$.

Proof:

By Prop. 11.3(c), $\underset{p \wedge q}{\sim}$ and $\underset{p \wedge \neg q}{\sim}$ are reflexive. Hence (i) follows from Prop. 11.2(i). Since $p \wedge q \wedge \neg q$ and $p \wedge \neg q \wedge q$ are contradictory, Prop. 11.3(d) implies $t' \underset{p \wedge q \wedge \neg q}{\sim} t$ and $t \underset{p \wedge \neg q \wedge q}{\sim} t'$. Thus $IF(q,t,t') \underset{p \wedge q}{\sim} t$ and $IF(q,t,t') \underset{p \wedge \neg q}{\sim} t'$ by Prop. 11.2(i) and reflexivity of $\underset{p \wedge q \wedge q}{\sim}$ and $\underset{p \wedge \neg q \wedge \neg q}{\sim}$, respectively. □

11.13 Lemma

Let $t,t' \in T$ such that

$$t = IF_1(p,IF_2(q,t_1,t_2),IF_2(q,t_3,t_4))$$

and

$$t' = IF_2(q,IF_1(p,t_1,t_3),IF_1(p,t_2,t_4))$$

for some $IF_1,IF_2 \in C$, $p,q \in BP$ and $t_1,\ldots,t_4 \in T$. Then t and t' are conditionally equal.

Proof:

By Prop. 11.2/3,

$$
\begin{aligned}
& t \underset{TRUE}{\sim} t' \\
\text{iff}\quad & IF_2(q,t_1,t_2) \underset{p}{\sim} t' \text{ and } IF_2(q,t_3,t_4) \underset{\neg p}{\sim} t' \\
\text{iff}\quad & t_1 \underset{p\wedge q}{\sim} t',\ t_2 \underset{p\wedge\neg q}{\sim} t',\ t_3 \underset{\neg p\wedge q}{\sim} t' \text{ and } t_4 \underset{\neg p\wedge\neg q}{\sim} t' \\
\text{iff}\quad & t' \underset{q\wedge p}{\sim} t_1,\ t' \underset{\neg q\wedge p}{\sim} t_2,\ t' \underset{q\wedge\neg p}{\sim} t_3 \text{ and } t' \underset{\neg q\wedge\neg p}{\sim} t_4 \\
\text{iff}\quad & IF_1(p,t_1,t_3) \underset{q\wedge p\wedge q}{\sim} t_1,\ IF_1(p,t_2,t_4) \underset{q\wedge p\wedge\neg q}{\sim} t_1, \\
& IF_1(p,t_1,t_3) \underset{\neg q\wedge p\wedge q}{\sim} t_2,\ IF_1(p,t_2,t_4) \underset{\neg q\wedge p\wedge\neg q}{\sim} t_2, \\
& IF_1(p,t_1,t_3) \underset{q\wedge\neg p\wedge q}{\sim} t_3,\ IF_1(p,t_2,t_4) \underset{q\wedge\neg p\wedge\neg q}{\sim} t_3, \\
& IF_1(p,t_1,t_3) \underset{\neg q\wedge\neg p\wedge q}{\sim} t_4 \text{ and } IF_1(p,t_2,t_4) \underset{\neg q\wedge\neg p\wedge\neg q}{\sim} t_4 \\
\text{iff}\quad & IF_1(p,t_1,t_3) \underset{q\wedge p}{\sim} t_1,\ IF_1(p,t_2,t_4) \underset{\neg q\wedge p}{\sim} t_2, \\
& IF_1(p,t_1,t_3) \underset{q\wedge\neg p}{\sim} t_3 \text{ and } IF_1(p,t_2,t_4) \underset{\neg q\wedge\neg p}{\sim} t_4.
\end{aligned}
$$

By Lemma 11.13, the last equivalent statement holds true. Hence $t \sim t'$. □

11.14 Lemma

Let $t,t' \in T$ with

$$t = IF(q,t_1,IF(p,t_2,t_3)),$$

$$t' = IF(q',IF(q,t_1,t_3),IF(p,t_2,IF(q,t_1,t_3)))$$

for some IF$\in$C, p,q,q'$\in$BP and $t_1, t_2, t_3 \in$ T such that $p \wedge q \wedge \neg q'$ as well as $p \wedge \neg q \wedge q'$ are contradictory. Then t and t' are conditionally equal.

<u>Proof:</u>

By assumption and Prop. 11.2/3,

$$t \underset{TRUE}{\sim} t'$$

iff $t_1 \underset{q}{\sim} t'$ and $IF(p,t_2,t_3) \underset{\neg q}{\sim} t'$
iff $t_1 \underset{q}{\sim} t'$, $t_2 \underset{\neg q \wedge p}{\sim} t'$ and $t_3 \underset{\neg q \wedge \neg p}{\sim} t'$
iff $t' \underset{q}{\sim} t_1$, $t' \underset{\neg q \wedge p}{\sim} t_2$ and $t' \underset{\neg q \wedge \neg p}{\sim} t_3$
iff $IF(q,t_1,t_3) \underset{q \wedge q'}{\sim} t_1$, $IF(p,t_2,IF(q,t_1,t_3)) \underset{q \wedge \neg q'}{\sim} t_1$,
$IF(q,t_1,t_3) \underset{\neg q \wedge p \wedge q'}{\sim} t_2$, $IF(p,t_2,IF(q,t_1,t_3)) \underset{\neg q \wedge p \wedge \neg q'}{\sim} t_2$,
$IF(q,t_1,t_3) \underset{\neg q \wedge \neg p \wedge q'}{\sim} t_3$ and
$IF(p,t_2,IF(q,t_1,t_3)) \underset{\neg q \wedge \neg p \wedge \neg q'}{\sim} t_3$
iff $IF(q,t_1,t_3) \underset{q' \wedge q}{\sim} t_1$, $t_2 \underset{q \wedge \neg q' \wedge p}{\sim} t_1$, $IF(q,t_1,t_3) \underset{q \wedge \neg q' \wedge \neg p}{\sim} t_1$,
$IF(p,t_2,IF(q,t_1,t_3)) \underset{\neg q \wedge \neg q' \wedge p}{\sim} t_2$, $IF(q,t_1,t_3) \underset{\neg p \wedge q' \wedge \neg q}{\sim} t_3$,
$t_2 \underset{\neg q \wedge \neg p \wedge \neg q' \wedge p}{\sim} t_3$ and $IF(q,t_1,t_3) \underset{\neg q \wedge \neg p \wedge \neg q' \wedge \neg p}{\sim} t_3$
iff $IF(q,t_1,t_3) \underset{q' \wedge q}{\sim} t_1$, $IF(q,t_1,t_3) \underset{\neg q' \wedge p \wedge q}{\sim} t_1$,
$IF(p,t_2,IF(q,t_1,t_3)) \underset{\neg q \wedge \neg q' \wedge p}{\sim} t_2$, $IF(q,t_1,t_3) \underset{\neg p \wedge q' \wedge \neg q}{\sim} t_3$
and $IF(q,t_1,t_3) \underset{\neg p \wedge \neg q' \wedge \neg q}{\sim} t_3$.

By Lemma 11.13, the last equivalent statement holds true. Hence $t \sim t'$. □

11.15 <u>Example</u> (arrayl)

Let BPAR = <entry,array> (cf. 1.5), PAR = <entry,arrayl> (cf. 2.2), K be the class of entry-algebras defined in 1.11 and C = {IFE,IFA}. We have shown in Ex. 8.6 that for all A$\in$K BSPEC(A) is a correct extension of bool.

By Thm. 11.11, PAR is consistent w.r.t.<BPAR,K>:

Set R = {a5, a6} (cf. 2.2). Clearly, R is linear.
By Ex. 8.6, R is directly decreasing and base-total w.r.t. BOP-C. Thus 11.11(i) follows from E-BE = R.
Since E-R' = E-(R∪{e1, e2, a3, a4}) (cf. 1.5), 11.11(ii) holds true. All simple critical pairs <t,t'> of <R,R> satisfy t = t'. Hence SCRIT(R,R) is absolutely R-convergent w.r.t. conditional equality. (Note that by Prop.11.3 (c), $\sim$ is reflexive.)

Let <t,t'>∈CRIT(R,E-R') with generator <<l,r>,h>.
Then <l,r> = a6, and

(a) a1 overlaps a6, i.e.

t = IFE(EQN(n,hm),UNDEF,GET(NEW,hm)),

t' = GET(NEW,hm)

or

(b) a2 overlaps a6, i.e.

t = IFE(EQN(m,hm),y,GET(PUT(a,n,x),hm)),

t' = GET(IFA(EQN(n,m),
PUT(a,m,y),
PUT(PUT(a,m,y),n,x),
hm).

Case (a) implies

$t \xrightarrow[\{a5\}]{}$ IFE(EQN(n,hm),UNDEF,UNDEF) =: u

and

$t' \xrightarrow[\{a5\}]{}$ UNDEF =:u'.

By Lemma 11.13, u$\sim$u'.

Case (b) yields

$t \xrightarrow[\{a6\}]{}$ IFE(EQN(m,hm),y,IFE(EQN(n,hm),x,GET(a,hm))) =: u

and

$$t' \xrightarrow[CCR]{} IFE(EQN(n,m), GET(PUT(a,m,y),hm), GET(PUT(PUT(a,m,y),n,x),hm))$$

$$\xrightarrow[\{\underline{a6}\}]{*} IFE(EQN(n,m), IFE(EQN(m,hm),y,GET(a,hm)), IFE(EQN(n,hm),x, GET(PUT(a,m,y),hm)))$$

$$\xrightarrow[\{\underline{a6}\}]{} IFE(EQN(n,m), IFE(EQN(m,hm),y,GET(a,hm)), IFE(EQN(n,hm),x,IFE(EQN(m,hm), y, GET(a,hm))))$$

$$=: u'.$$

Let $p = EQN(n,hm)$, $q = EQN(m,hm)$, $q' = EQN(n,m)$, $A \in K$ and $f \in BZ(G(A))$. By definition of EQN_A (cf. 1.11),

$$f(p \wedge q)_A = TRUE_A \text{ implies } (fq')_A = TRUE_A$$

and

$$f(p \wedge q')_A = TRUE_A \text{ implies } (fq)_A = TRUE_A.$$

Hence $p \wedge q \wedge \neg q'$ and $p \wedge \neg q \wedge q'$ are contradictory so that by Lemma 11.14, $u \sim u'$.
Therefore, $<t,t'>$ is absolutely (and thus relatively) $(R \cup CCR)$-convergent w.r.t. $\sim$ in both cases.
Clearly, $<t,t'>$ is not including.
Since for all $<l,r> \in E-R'$ $GET \notin op(l)$, $CRIT(E-R',R)$ is empty.
For all $<l,r> \in CCR$

$$l = \sigma(v, IF(b,x,y), w)$$

and

$$r = IF(b, \sigma(v,x,w), \sigma(v,y,w))$$

for some $\sigma \in \{EQE, PUT, GET\}$, $IF \in \{IFE, IFA\}$, $b,x,y \in X$ and $v,w \in X^*$, while $R \cup (E-R') = E-\{\underline{e1},\underline{e2},\underline{a3},\underline{a4}\}$.
Hence $CRIT(CCR, R \cup (E-R'))$ is empty, too. □

11.16 Example (setl)

Let BPAR = <entry,set>, PAR = <entry, setl> (cf. 8.7), K be the class of entry-algebras defined in 1.11 and C = {IFS}. Set R = {s5, s6}. Analogously to Ex. 11.15, one obtains all assumptions of Thm. 11.11 except for relative convergence of CRIT(R,E-R'). So let $<t,t'> \in$ CRIT(R,E-R') with generator <<l,r>,h>.

Then <l,r> = s6 and

(a) s1 overlaps s6, i.e.

t = IFS(EQE(x,hy),DEL(INS(s,x),hy),
INS(DEL(INS(s,x),hy),x)),

t' = DEL(INS(s,x),hy)

or

(b) s2 overlaps s6, i.e.

t = IFS(EQE(y,hy),DEL(INS(s,x),hy),
INS(DEL(INS(s,x),hy),y)),

t' = DEL(INS(INS(s,y),x),hy).

Let p = EQE(x,hy),q = EQE(y,hy) and t_o = DEL(s,hy).

Case (a) implies

$$
\begin{aligned}
t &\xrightarrow[\{s6\}]{*} \mathrm{IFS}(p,\mathrm{IFS}(p,t_o,\mathrm{INS}(t_o,x)),\\
&\qquad \mathrm{INS}(\mathrm{IFS}(p,t_o,\mathrm{INS}(t_o,x)),x))\\
&\xrightarrow[CCR]{} \mathrm{IFS}(p,\mathrm{IFS}(p,t_o,\mathrm{INS}(t_o,x)),\\
&\qquad \mathrm{IFS}(p,\mathrm{INS}(t_o,x),\mathrm{INS}(\mathrm{INS}(t_o,x),x)))\\
&\xrightarrow[\{s1\}]{} \mathrm{IFS}(p,\mathrm{IFS}(p,t_o,\mathrm{INS}(t_o,x)),\\
&\qquad \mathrm{IFS}(p,\mathrm{INS}(t_o,x),\mathrm{INS}(t_o,x)))\\
&=: u
\end{aligned}
$$

and

$$t' \xrightarrow[\{s6\}]{} IFS(p,t_o,INS(t_o,x)) =: u'.$$

By Prop. 11.2/3,

$$u \underset{TRUE}{\sim} u'$$

$$\begin{aligned} &\text{iff}\quad u' \underset{p}{\sim} u' \text{ and } IFS(p,INS(t_o,x),INS(t_o,x)) \underset{\neg p}{\sim} u' \\ &\text{iff}\quad u' \underset{\neg p}{\sim} IFS(p,INS(t_o,x),INS(t_o,x)) \\ &\text{iff}\quad t_o \underset{\neg p \wedge p}{\sim} IFS(p,INS(t_o,x),INS(t_o,x)) \\ &\qquad \text{and } INS(t_o,x) \underset{\neg p}{\sim} IFS(p,INS(t_o,x),INS(t_o,x)) \\ &\text{iff}\quad INS(t_o,x) \underset{\neg p}{\sim} IFS(p,INS(t_o,x),INS(t_o,x)). \end{aligned}$$

By Lemma 11.12 , the last equivalent statement holds true. Hence $u \sim u'$.

Case (b) implies

$$\begin{aligned} t &\xrightarrow[\{s6\}]{*} IFS(q,IFS(p,t_o,INS(t_o,x)), \\ &\qquad\qquad INS(IFS(p,t_o,INS(t_o,x)),y)) \\ &\xrightarrow[CCR]{} IFS(q,IFS(p,t_o,INS(t_o,x)), \\ &\qquad\qquad IFS(p,INS(t_o,y),INS(INS(t_o,x),y))) \\ &\xrightarrow[\{s2\}]{} IFS(q,IFS(p,t_o,INS(t_o,x)), \\ &\qquad\qquad IFS(p,INS(t_o,y),INS(INS(t_o,y),x))) \\ &\qquad =: u \end{aligned}$$

and

$$\begin{aligned} t' &\xrightarrow[\{s6\}]{} IFS(p,DEL(INS(s,y),hy), \\ &\qquad\qquad INS(DEL(INS(s,y),hy),x)) \\ &\xrightarrow[\{s6\}]{*} IFS(p,IFS(q,t_o,INS(t_o,y)), \\ &\qquad\qquad INS(IFS(q,t_o,INS(t_o,y)),x)) \\ &\xrightarrow[CCR]{} IFS(p,IFS(q,t_o,INS(t_o,y)), \\ &\qquad\qquad IFS(q,INS(t_o,x),INS(INS(t_o,y),x))) \\ &\qquad =: u'. \end{aligned}$$

By Lemma 11.13 , $u \sim u'$.

Thus in both cases, $\langle t,t' \rangle$ is relatively (but not absolutely) $(R \cup CCR)$-convergent w.r.t. $\sim$. $\square$

Conclusion: Some remarks on the history of this work

When we (Hartmut Ehrig, Hans-Jörg Kreowski and the author) started to do research on algebraic specifications of abstract data types in 1977, we soon came across the verification problem of data type extensions. So we devoted chapter 2 of Ehrig, Kreowski, Padawitz /18/ to the "stepwise specification by enrichment" that culminated in an "enrichment theorem for inductively specified operations". We introduced two criteria which should imply completeness: "generating" and "weight-decreasing" equations, and two for consistency: "unequivocal" and "BE-consistent" equations. But Bernhard Josko /36/ gave a counter-example to our conjecture that generating and weight-decreasing equations are sufficient completeness conditions. The crucial point is that our notion of "generating" was too weak: "generating" could be called "base-total modulo BPAR". (Replace "ft = u" in Def. 7.1 by "$ft \equiv_{BSPEC(A)} u$ for all $A \in K$".) Ehrig, Kreowski, Padawitz /19/ provides a corrected version of the completeness criteria by defining "generating" as "base-total modulo BPAR" together with an additional syntactical requirement.

More serious was the fact that the consistency criteria of Ehrig, Kreowski, Padawitz /18/ were too strong even for the int-specification of Ex. 9.19. They bear the same drawback which applies to "Huet's" Consistency Theorem 10.3, namely that BE is regarded as a symmetric relation. Moreover, we have to assume the linearity of BE, BE^{-1} and E-BE. Otherwise unequivocal and BE-consistent equations (of E-BE) in general do not imply consistency.

These difficulties with adhoc solutions of the extension problem as well as the consideration of parameterized specifications suggested to lay down a general framework for a variety of correctness, completeness and consistency criteria fitting different situations. Prior to our first attempts to tackle the correctness problem were two approaches I did not include into my general framework. One of them uses "canonical term algebras" as mediators between a model and its specification. Advocates of this approach were ADJ/1/ (5.4.1), Nourani/50/, Klären /40/ and Veloso /61/. In some cases canonical term algebras provide vivid correctness proofs, but in others they do not arise as abstract models and thus became unhandy for correctness proofs. On the other hand, their basic idea of assigning a unique term to each element of the model appears in all correctness proofs (cf. Thm. 1.15).

The other approach to correctness criteria came up with Guttag /24/ and Guttag, Horning /25/ and is concerned with the case that PSPEC equals BSPEC: I noted that Guttag's sufficient completeness and consistency agree with the characterizations of completeness resp. consistency given in Prop. 2.19. Guttag, Horning /25/ present a syntactical criterion for sufficient completeness, which is weak enough to allow axiomatizations of all primitive-recursive functions.

In contrast to these special-purpose approaches the theory of term rewriting systems developped by Rosen /60/ and O'Donnell /52/ has turned out be more appropriate for a proof theory of algebraic specifications. As far as I know Wand /62/ was the first one who displayed the connection between equational theories and term rewriting. Raoult, Vuillemin /58/ are mainly concerned with term rewriting in the context of program semantics, but their proposition 10 is an elegant reformulation of Rosen's /60/ Rule-Schemata Thm. 6.5, which is needed in my Consistency Thm. 4.19. Huet's and Hullot's /32/ definition of complete term tupel sets provide an idea for the decidability of base-completeness (Corollary 7.5). The original definitions of simplification and recursive path orderings were given by Dershowitz /11/, /12/ and Plaisted /57/. The significance of Huet /30/ for our notion of relative confluence was discussed in 10.1-10.4. Finally, extensions of the Knuth-Bendix algorithm to theorem provers for equational theories (Musser /48/, Goguen /23/, Huet, Hullot /32/) were investigated in section 3.2.

Padawitz /53/ was my first approach to completeness and consistency criteria which are build upon results in the theory of term rewriting systems. Chapters 9 and 10 of the present thesis can be regarded as a further development of Padawitz /53/. All other results presented here, the adaption to parameterized specifications and the treatment of specifications with conditionals, were not published before.

I would like to thank Hartmut Ehrig and Dirk Siefkes for supervising this thesis and for many fruitful discussions on parameterized specifications. I am indebted to Werner Fey for inspiring talks about his parts-system specification, and I owe thanks to Gerard Huet who provided valuable comments on Padawitz /53/, which influenced the further development of that paper.

References

/1/ ADJ: J.A. Goguen, J.W. Thatcher, E.G. Wagner, An Initial Algebra Approach to the Specification, Correctness and Implementation of Abstract Data Types, in: R.T.Yeh, ed., Current Trends in Programming Methodology, Vol. 4, Prentice-Hall (1978) 80-149.

/2/ ADJ: J.W. Thatcher, E.G. Wagner, J.B. Wright, Data Type Specification: Parameterization and the Power of Specification Techniques, ACM Transactions on Programming Languages and Systems 4 (1982) 711-732.

/3/ ADJ: H. Ehrig, H.-J. Kreowski, J.W. Thatcher, E.G. Wagner, J.B. Wright, Parameter Passing in Algebraic Specification Languages, Proc. Workshop on Program Specification, Springer Lecture Notes in Computer Science 134 (1982) 322-369.

/4/ M.A. Arbib, E.G. Manes, Arrows, Structures and Functors, Academic Press (1975).

/5/ J. de Bakker, Mathematical Theory of Program Correctness, Prentice-Hall (1980).

/6/ M. Bidoit, Proofs by Induction in "Fairly" Specified Equational Theories, 6th German Workshop on Artificial Intelligence, Springer Informatik-Fachberichte 58 (1982) 154-166.

/7/ R.S. Boyer, J.S. Moore, A Computational Logic, Academic Press (1979).

/8/ W.S. Brainerd, L.H. Landweber, Theory of Computation, John Wiley & Sons (1974).

/9/ P.M. Cohn, Universal Algebra, D. Reidel Publishing Company (1981).

/10/ O.-J. Dahl, C.A.R. Hoare, Hierarchical Program Structures, in: O.-J. Dahl, E.W. Dijkstra, C.A.R. Hoare, Structured Programming, Academic Press (1972) 175-220.

/11/ N. Dershowitz, A Note on Simplification Orderings, Information Processing Letters 9 (1979) 212-215.

/12/ N. Dershowitz, Orderings for Term Rewriting Systems, Theoretical Computer Science 17 (1982) 279-301.

/13/ N. Dershowitz, Z. Manna, Proving Termination with Multiset Orderings, Communications ACM 22 (1979) 465-476.

/14/ E.W. Dijkstra, Notes on Structured Programming, in: O.-J. Dahl, E.W. Dijkstra, C.A.R. Hoare, Structured Programming, Academic Press (1972) 1-82.

/15/ H.-D. Ehrich, On the Theory of Specifications, Implementation and Parameterization of Abstract Data Types, Journal ACM 29 (1982) 206-227.

/16/ H. Ehrig, Algebraic Theory of Parameterized Specifications with Requirements, Proc. 6th Colloquium on Trees in Algebra and Programming, Springer Lecture Notes in Computer Science 112 (1981) 1-24.

/17/ H. Ehrig, W. Fey, Methodology for the Specification of Software Systems: From Formal Requirements to Algebraic Design Specifications, Proc. GI-11. Jahrestagung, Springer Informatik-Fachberichte 50 (1981) 255-269.

/18/ H. Ehrig, H.-J. Kreowski, P. Padawitz, Stepwise Specification and Implementation of Abstract Data Types, Proc. 5th Int. Colloquium on Automata, Languages, and Programming, Springer Lecture Notes in Computer Science 62 (1978) 205-226.

/19/ H. Ehrig, H.-J. Kreowski, P. Padawitz, Completeness in Algebraic Specifications, EATCS Bulletin 11 (1980) 2-9.

/20/ H. Ehrig, H.-J. Kreowski, B. Mahr, P. Padawitz, Algebraic Implementation of Abstract Data Types, Theoretical Computer Science 20 (1982) 209-264.

/21/ R.W. Floyd, Assigning Meanings to Programs, in: J.T. Schwartz, ed., Mathematical Aspects of Computer Science, American Mathematical Society (1967) 19-32.

/22/ H. Ganzinger, A Final Algebra Semantics for Parameterized Specifications, Technical Report, Technische Universität München (1980).

/22a/ V. Giarratana, F. Gimona, U. Montanari, Observability Concepts in Abstract Data Type Specification, Proc. 5th Symposium on Mathematical Foundations of Computer Science, Springer Lecture Notes in Computer Science 45 (1976) 575-587.

/23/ J.A. Goguen, How to Prove Algebraic Inductive Hypothesis without Induction, Proc. 5th Conference on Automated Deduction, Springer Lecture Notes in Computer Science 87 (1980) 356-373.

/24/ J.V. Guttag, The Specification and Application to Programming of Abstract Data Types, Ph. D. Thesis, Department of Computer Science, University of Toronto (1975)

/25/ J.V. Guttag, J.J. Horning, The Algebraic Specification of Abstract Data Types, Acta Informatica 10 (1978) 27-52

/26/ J.V. Guttag, E. Horowitz, D.R. Musser, Abstract Data Types and Software Validation, Communications ACM 21 (1978) 1048-1064.

/27/ C.A.R. Hoare, An Axiomatic Basis for Computer Programming, Communications ACM 12 (1969) 576-580.

/28/ C.A.R. Hoare, Proof of Correctness of Data Representations, Acta Informatica 1 (1972) 271-281.

/29/ G. Hornung, P. Raulefs, Terminal Algebra Semantics and Retractions for Abstract Data Types, Proc. 7th Int. Colloquium on Automata, Languages, and Programming, Springer Lecture Notes in Computer Science 85 (1980) 310-323.

/30/ G. Huet, Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems, Journal ACM 27 (1980) 797-821.

/31/ G. Huet, A Complete Proof of Correctness of the Knuth-Bendix Completion Algorithm, Journal of Computer and System Sciences 23 (1981) 11-21.

/32/ G. Huet, J.-M. Hullot, Proofs by Induction in Equational Theories with Constructors, Journal of Computer and System Sciences 25 (1982) 239-266.

/33/ G. Huet, D. Lankford, On the Uniform Halting Problem, Technical Report 283, INRIA Laboria (1978).

/34/ G. Huet, D.C. Oppen, Equations and Rewrite Rules: A Survey, in: R.V. Book, ed., Formal Language Theory: Perspectives and Open Problems, Academic Press (1980).

/35/ H.-J. Jeanrond, Deciding Unique Termination of Permutative Rewriting Systems: Choose Your Term Algebra Carefully, Proc. 5th Conference on Automated Deduction, Springer Lecture Notes in Computer Science 87 (1980) 335-355.

/36/ B. Josko, Das Enrichment-Problem, manuscript for a seminar (in German), Mathematisch-Naturwissenschaftliche Fakultät, Technische Hochschule Aachen (1979).

/37/ J.P. Jouannaud, Confluent and Coherent Sets of Reductions with Equations, Proc. 8th Colloquium on Trees in Algebra and Programming (1983).

/38/ S. Kamin, Final Data Type Specifications: A New Data Type Specification Method, Proc. 7th ACM Symposium on Principles of Programming Languages (1980) 131-138.

/39/ S. Kamin, J.-J. Levy, Attempts for Generalizing the Recursive Path Orderings, manuscript, INRIA Laboria (1980).

/40/ H.A. Klären, A Simple Class of Algorithmic Specifications for Abstract Software Modules, Proc. 9th Symposium on Mathematical Foundations of Computer Science, Springer Lecture Notes in Computer Science 88 (1980) 362-374.

/41/ H.A. Klären, H. Petzsch, The Development of an Interpreter by means of Abstract Algebraic Software Specifications, Proc. Int. Colloquium on Formalization of Programming Concepts, Springer Lecture Notes in Computer Science 107 (1981) 335-346.

/42/ D.E. Knuth, The Art of Computer Programming, Vol. 1: Fundamental Algorithms, Addison-Wesley Publishing Company (1968).

/43/ D.E. Knuth, P.B. Bendix, Simple Word Problems in Universal Algebras, in: J. Leech, ed., Computational Problems in Abstract Algebra, Pergamon Press (1970) 263-297.

/44/ J.B. Kruskal, Well-quasi-ordering, the Tree Theorem, and Vaszonyi's Conjecture, Transactions AMS 95 (1960) 210-225.

/45/ D.S. Lankford, A.M. Ballantyne, Decision Procedures for Simple Equational Theories with Permutative Axioms: Complete Sets of Permutative Reductions, Technical Report ATP-3, Department of Mathematics and Computer Science, University of Texas at Austin (1977).

/46/ B. Liskov, S. Zilles, Programming with Abstract Data Types, SIGPLAN Notices 9 (1974) 55-59.

/47/ Z. Manna, J. Vuillemin, Fixpoint Approach to the Theory of Computation, Communications ACM 15 (1972) 528-536.

/48/ D.L. Musser, On Proving Inductive Properties of Abstract Data Types, Proc. 7th ACM Symposium on Principles of Programming Languages (1980) 154-162.

/49/ T. Nipkow, G. Weikum, A Decidability Result about Sufficient Completeness of Axiomatically Specified Abstract Data Types, Proc. 6th GI-Conference on Theoretical Computer Science, Springer Lecture Notes in Computer Science 145 (1983) 257-268.

/50/ C.F. Nourani, Constructive Extension and Implementation of Abstract Data Types and Algorithms, Ph.D. Thesis, Computer Science Department, University of California at Los Angeles (1979).

/51/ C.F. Nourani, On Induction for Programming Logic: Syntax, Semantics, and Inductive Closure, Technical Report, Department of Computer and Information Science, University of Pennsylvania, Philadelphia (1981).

/52/ M.J. O'Donnell, Computing in Systems Described by Equations, Springer Lecture Notes in Computer Science 58 (1977).

/53/ P. Padawitz, New Results on Completeness and Consistency of Abstract Data Types, Proc. 9th Symposium on Mathematical Foundations of Computer Science, Springer Lecture Notes in Computer Science 88 (1980) 460-473.

/54/ D.L. Parnas, A Technique for Software Module Specification with Examples, Communications ACM 15 (1972) 330-336.

/55/ G.E. Peterson, M.E. Stickel, Complete Sets of Reductions for some Equational Theories, Journal ACM 28 (1981) 233-264.

/56/ A. Pettorossi, Comparing and Putting Together Recursive Path Ordering, Simplification Orderings and Non-Ascending Property for Termination Proofs of Term Rewriting Systems, Proc. 8th Int. Colloquium on Automata, Languages, and Programming, Springer Lecture Notes in Computer Science 115 (1981) 431-447.

/57/ D.A. Plaisted, A Recursively Defined Ordering for Proving Termination of Term Rewriting Systems, Technical Report R-78-943, Department of Computer Science, University of Illinois at Urbana-Champaign (1978).

/58/ J.-C. Raoult, J. Vuillemin, Operational and Semantic Equivalence Between Recursive Programs, Journal ACM 27 (1980) 772-796.

/59/ J.A. Robinson, A Machine-Oriented Logic Based on the Resolution Principle, Journal ACM 12 (1965) 23-41.

/60/ B.K. Rosen, Tree-Manipulating Systems and Church-Rosser Theorems, Journal ACM 20 (1973) 160-187.

/61/ P.A.S. Veloso, Methodical Specification of Abstract Data Types via Rewriting Systems, Technical Report 7/81, Departamento de Informática, Pontificia Universidade Católica do Rio de Janeiro (1981).

/62/ M. Wand, Algebraic Theories and Tree Rewriting Systems, Technical Report 66, Computer Science Department, Indiana University, Bloomington (1977).

/63/ M. Wand, A New Incompleteness Result for Hoare's System, Journal ACM 25 (1978) 168-175.

/64/ M. Wand, FinalAlgebra Semantics and Data Type Extensions, Journal of Computer and System Sciences 19 (1979) 27-44.

/65/ N. Wirth, The Development of Programs by Stepwise Refinement , Communications ACM 14 (1971) 221-227.

SUBJECT INDEX

AUTHOR INDEX